WebGUI Administrators Guide

by JT Smith and Kristi McCombs

Printed in the United States of America by Lulu.com.

Published by Plain Black Corporation, 1360 Regent St, #145, Madison, WI, 53715.

Editor: Kristi McCombs

Cover Design: Darci Gibson

Printing History: September 2007: First Edition

Preface

This book was written for you, the Administrator. You won't find yourself wading through a bunch of end user or programmer documentation. Instead, this book was carefully constructed with just the nuggets of knowledge you'll care about to get your job done as quickly, robustly, and securely as possible.

This book assumes that you have at least a good familiarity with Linux® or Unix®. Though there are some chapters that specifically mention Microsoft Windows®, the majority of WebGUI users, and therefore the majority of this book, use examples from a *nix environment. However, almost everything should translated neatly from *nix into Windows; for example, if you see */data*, then it would be *c:\data* in Windows.

Table of Contents

Admin Console Quick Reference

- Active Sessions: This will display all sessions currently running on your site. It shows user id, session signature, when it will expire from inactivity, time of the last page view, IP address, and there is a button for you to administratively kill the session.
- Advertising: A complete ad management system that can be used for placing advertisements on your site, tracking clicks and impressions, or just rotating random content on your site.
- Assets: This will take you to the asset manager. Here you can see all assets, where they rank, the type of asset, and how large it is. You can also edit and add assets.
- Cache: Displays the current cache usage, and allows you to clear the cache.

- Clipboard: This is where any cut, copied, or shortcutted content resides.
- Commerce: This is where you can set up all commerce functions, from checkout and transaction templates to setting up payment and shipping plugins.
- Content Filters: This feature allows censoring of the site. You can use it to stop inappropriate language or to stop the use of bad code or macros that may cause damage to your site.
- Content Profiling: Allows you to create metadata fields to apply to assets throughout the site.
- Databases: This section shows those databases that have been linked to WebGUI and are available to be queried.
- Graphics: Allows you to set up color pallets and fonts to be used in graphing and graphics functions in the site.
- Groups: This allows you to create and manage groups. You can create and delete groups, add groups to existing groups and see the users that are part of a particular group. Here is where you can set up users being automatically added and removed from groups.
- Help: This takes you to the main page of the help index, which provides a link to the help wiki and provides a listing of template variables available in your system.
- Inbox: This is your inbox where WebGUI will place actions that you need to perform, private messages from other users, and system notices. If you have an email address defined in your profile, all messages sent here will also be forwarded to your email address.
- LDAP Connections: This is where you can set up and maintain LDAP connections to WebGUI.
- Login History: This displays the login history of everyone who's logged into your site. It shows the user id, whether the login was successful or not, the time, the IP address, and the browser used. This is especially useful to track those who may be trying to illegally access the site.
- Products: This contains a list products for sale through the commerce system. Through this feature new products can be

created, and existing products managed.

- Scheduler: This allows you to schedule specific tasks to be performed on the site at a specific date and time. This is commonly used in conjunction with workflows.

- Settings: This allows the set up of general WebGUI settings. It allows entry of company information, setting of content sizes, starting UI levels, and several other authentication options.

- Spectre: This screen shows you whether S.P.E.C.T.R.E.,WebGUI's workflow governor, is running and how busy it is.

- Statistics: This shows the basic statistics of your WebGUI site such as WebGUI version number, number of assets, number of templates, active sessions, groups, and users.

- Subscriptions: This allows set up of subscriptions you can sell within your WebGUI site.

- Trash: Trash contains all those pieces of content that were deleted from the site. It is here that recovery of accidentally deleted pieces of content takes place.

- User Profiling: This is where the user profile fields are added and deleted to allow customization of the information required of users when new user ids are created.

- Users: This is where all user information is kept. A search is available to narrow the number of users returned. It shows users' names, status, as well as when they were created, among other things. Here you can add users, manage groups, and even become a particular user.

- Version Tags: This screen lists all currently active version tags in the system. You can edit existing tags and create new tags. In this screen you can also manage pending and committed versions, as well as roll back mistakenly committed version tags.

- Workflow: Workflows handle the movement of content and information through the site. The workflow feature allows you to create customized workflows to be performed on the site.

Installation

Installing WebGUI is the most difficult task you'll ever have to do with WebGUI. This is not because WebGUI is difficult to install, but rather that it has a lot of prerequisites. You have three options for this process:

- Easy: Host WebGUI with a professional hosting provider that will install it for you.
- Medium: Install using the WebGUI Runtime Environment.
- Difficult: Install WebGUI from source.

WebGUI Hosting Provider

The easiest way to install WebGUI is to have someone else install it! There are WebGUI hosting providers all over the world, and using their services is often the fastest and easiest way to get WebGUI installed. WebGUI's creator, Plain Black Corporation, also provides hosting services for WebGUI. You can learn more about Plain Black's hosting services at www.plainblack.com/hosting

Installation Using the WRE

The WRE, or WebGUI Runtime Environment, is the easiest way to install WebGUI if you want to install WebGUI on your own hardware. In addition to being the easiest way to install WebGUI yourself, the WRE also provides a number of utilities and performance enhancements to make administering WebGUI easier.

Download

Download the latest WRE from www.getwebgui.com.

Platform Specifics

Now you need to prepare your operating system for the WRE. The following sections provide instructions for common systems.

All *nix Operating Systems

If you have installed the MySQL package for your system you'll need to uninstall it, or at the very least disable it and remove /etc/my.cnf, as it will interfere with the MySQL that comes with the WRE. You can run a separate MySQL if you wish. Just put it on a different port and move the my.cnf into the data folder that came with your MySQL distribution, which is often either /var/lib/mysql or /usr/local/mysql/data.

If you have installed the Apache package for your system, you'll need to uninstall it, disable it, put it on a different port, or when prompted by the WRE, run modproxy on a port other than port 80.

You need to add a user to the operating system for the WRE to run under. We recommend a user named “webgui”. On most platforms you can run one of the following commands to add the user:

```
adduser -s /sbin/nologin webgui

useradd -s /sbin/nologin webgui
```

You can use an existing user on your server like “nobody” for this purpose, but it's less secure to do so since other services may be using it as well.

Linux

A default server install is all that is necessary to run the WRE on Red Hat Enterprise Linux, Ubuntu, Debian, and CentOS. No further action necessary.

Mac OS X Tiger (10.4)

You will need to install the X Code developer tools from Apple. You can download them from http://connect.apple.com.

FreeBSD

From the Ports system you'll need to install the “bash” and “libiconv” packages. You'll also need to run the following command:

```
ln -s /usr/local/bin/bash /bin/bash
```

Microsoft® Windows®

If you have IIS running on port 80, you'll either need to disable it, or choose a port other than port 80 for mod_proxy during the WRE installation

process.

The WRE installs on drive C:, so you'll need to have at least 400MB available to install the WRE, plus whatever space you need for sites. Note that it is possible after the install to put site data on an alternate drive.

You'll also need to download and install a couple of utilities. First, you'll need an archive extractor, like 7-Zip or WinZip, that can extract tarballs. You'll also need a good text editor (Notepad and Wordpad don't count). We recommend Notepad++.

All Platforms

You should always read the docs/install.txt file that comes with the WRE in case there are any version specific changes of which you should be aware.

Extracting The Tarball

Use your archiver to extract the WRE tarball. If you have GNU tar or equivalent, use the following command to extract it:

```
mkdir /data
cd /data
tar xvfz /path/to/wre-0.8.0-rhel4-ia32.tar.gz
```

Initial Configuration

To configure the WRE you need to use the WRE Console. Use the following command to start it:

```
/data/wre/sbin/wreconsole.pl
```

It will output the following:

```
Please contact me at:
        http://localhost.localdomain:60834/
```

However, for initial configuration you need to add /setup to the end of that URL (http://localhost.localdomain:60834/setup). Note that you can also access the WRE Console from an IP address or hostname bound to the machine, though you may have to open port 60834 in the firewall.

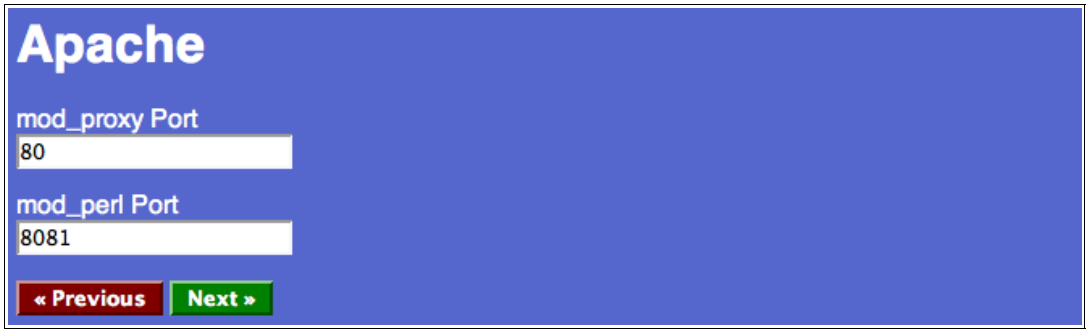
Although other browsers may work, this process is designed for Firefox and Internet Explorer compatibility.

You will then be lead through a series of screens with questions for you to answer. In most cases you can simply use the defaults. If you make a mistake, don't worry. You can always go backward in the process and change your answers, until you come to the last screen.

Specify your WRE operating system user.

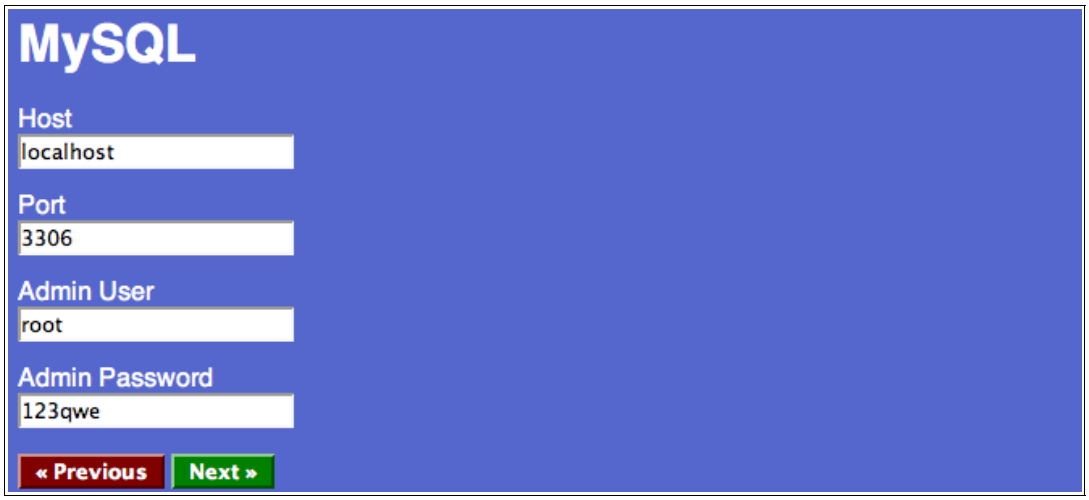
If you're not on Winows®, you'll also have the option of setting up a development only environment. This automatically configures mod_perl with Apache2::Reload so that you don't need to restart Apache every time you make a change to one of your WebGUI plugins. In addition, it will set up a dev site for you to work from.

Now, you can specify the ports for mod_proxy and mod_perl. If you're already running a web server on port 80, then you'll want to change the mod_proxy port to something else, perhaps 8080. Otherwise, you can safely leave the defaults.



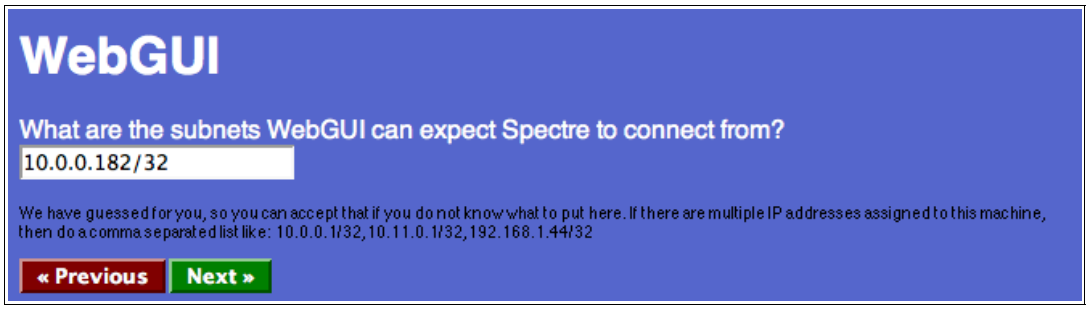
Next, specify information about how you'd like MySQL configured. If you already have MySQL configured on another server and you'd like to use that, then simply specify the IP or hostname of the other server, its port, and a username and password that have full administrative privileges, and

the WRE will configure itself to use that server.



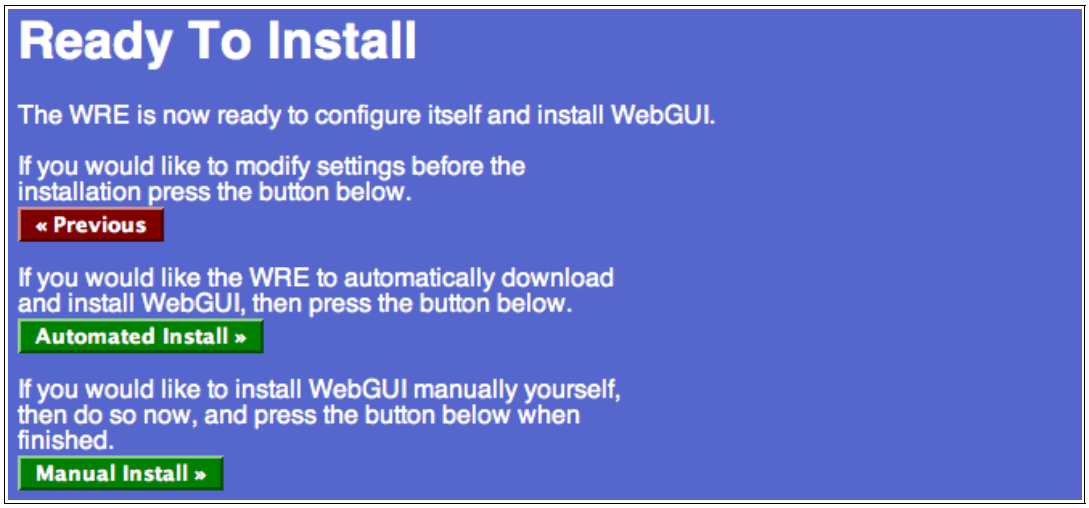
Finally, you need to tell the WRE what subnet Spectre will use to communicate with WebGUI. The WRE tries to guess this information and puts its guess in the box for you. However, if it guesses incorrectly, or you'll be running Spectre on a different machine, or you have multiple IP addresses bound to this machine, you'll need to specify those changes here.

Note that the WRE accepts these IP addresses using CIDR notation. Therefore, if you want WebGUI to be able to accept requests from all IP addresses in a range, like 10.1.1.0 – 10.1.1.255, you can specify that like “10.1.1.0/24”. Or, if you have multiple IP's, you can specify them with a comma separated list like “10.1.1.1/32,192.168.0.11/32”.

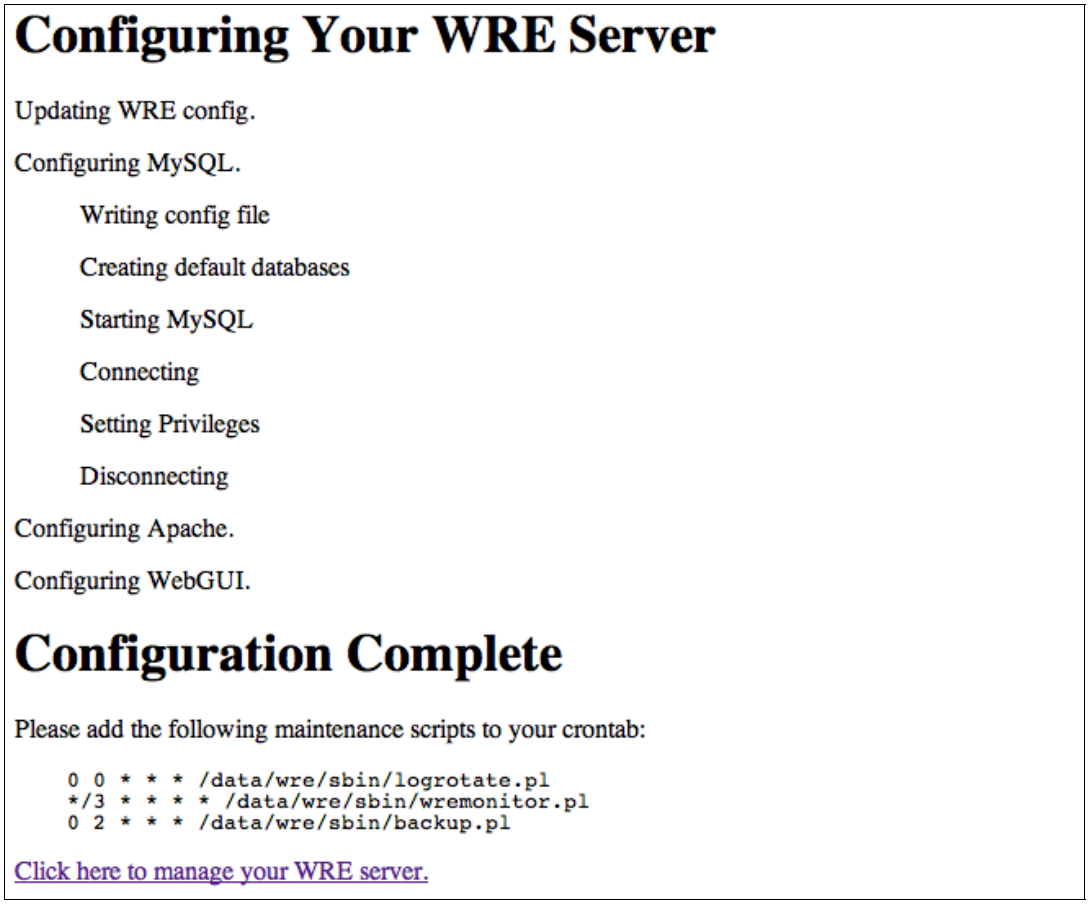


The WRE is now ready to apply all the configuration options you gave it. From this screen you can either go back and change your answers, or you can do an automated install which will automatically download and configure WebGUI. You can also do a manual install, which assumes that

you have already downloaded WebGUI and extracted it into /data/WebGUI. In most cases you want to choose “Automated Install”.



Depending on what options you chose in the previous screens, you should see a screen similar to the following.



Configuring Your WRE Server

Updating WRE config.

Configuring MySQL.

- Writing config file
- Creating default databases
- Starting MySQL
- Connecting
- Setting Privileges
- Disconnecting

Configuring Apache.

Configuring WebGUI.

Configuration Complete

Please add the following maintenance scripts to your crontab:

```
0 0 * * * /data/wre/sbin/logrotate.pl
*/3 * * * * /data/wre/sbin/wremonitor.pl
0 2 * * * /data/wre/sbin/backup.pl
```

Click here to manage your WRE server.

If any errors occur during this process do not continue. Instead, shut down the local MySQL server, delete /data/wre/var/mysqldata and /data/wre/etc/wre.conf, correct the problem, and then restart the initial configuration process.

Set Up Cron Jobs

The last step in the configuration process is to set up the cron jobs as shown on your configuration screen. You can usually do this by typing:

```
crontab -e
```

And then add the lines like this:

```
0 0 * * * /data/wre/sbin/logrotate.pl

*/3 * * * * /data/wre/sbin/wremonitor.pl

0 2 * * * /data/wre/sbin/backup.pl
```

WRE Install Complete

You have now completed the WRE and WebGUI installation, but before you can use WebGUI you need to add a site. That is, unless you chose the development only option during the configuration process. **To add a site skip to the next chapter.**

Source Install

Instead of using the WRE, you can do what is known as a source install. It needs to be said that this process is difficult and time consuming, even for experienced system administrators. With the exception of a strong desire to just “do-it-yourself”, you should always use the WRE instead of opting for this installation method.

That said, if you do decide to proceed, you'll need to manually install and configure all of WebGUI's pre-requisites. The instructions for doing so are contained in the remainder of this chapter. For most of the prerequisites we have not provided detailed build instructions, because you'll likely be using the package management system on your operating system to install these packages.

Perl

WebGUI requires Perl 5.8 or higher, which you can download from www.perl.org, or you can likely install from your operating system's package management system. Be careful, though, because a lot of operating systems still distribute old versions of Perl instead of Perl 5.8.

Graphics Magick

WebGUI requires Graphics Magick 1.1 or higher, which you can downbad from www.graphicsmagick.org, or you may be able to install using your operating system's package management system.

Also be sure to install the Perl Magick plugin that comes with Graphics Magick, and install the following image processing libraries:

- libpng - www.libpng.org
- libungif – sourceforge.net/projects/libungif
- libjpeg - www.ijg.org

- freetype - www.freetype.org

For WebGUI 7.4 and below you can also use Image Magick (instead of Graphics Magick), which is sometimes easier to come by in native packages.

MySQL

WebGUI requires MySQL 5.0 or higher, which you can download from www.mysql.com, or you can likely install from your operating system's package management system. Be careful, though, because some operating systems still distribute old versions of MySQL instead of MySQL 5.0.

Apache

WebGUI requires Apache 2.0 or higher with mod_perl installed. You can download Apache from httpd.apache.org and mod_perl from perl.apache.org, or you can likely install both of them using your operating system's package management system. Be careful, because a lot of operating systems are still distributing Apache 1.3 instead of Apache 2.0.

You'll also need to install libapreq2, which WebGUI needs to better communicate with Apache. You can download this from httpd.apache.org/apreq, or install from your operating system's package manager.

If you want a secure web site you'll also want to download and install OpenSSL 0.9.7m or higher from www.openssl.org, and you can certainly install this from your operating system's package manager.

Now, you'll need to edit your httpd.conf to include the following directives. These should be placed at the end of the module loading section of your config file.

```
LoadModule apreq_module modules/mod_apreq2.so

LoadModule perl_module modules/mod_perl.so

PerlSetVar WebguiRoot /data/WebGUI

PerlRequire /etc/modperl.pl

PerlCleanupHandler Apache2::SizeLimit
```

```
PerlRequire /data/WebGUI/sbin/preload.perl

Alias /extras /data/WebGUI/www/extras
```

Now create a file called /etc/modperl.pl and enter the following contents into it:

```
use Apache2::SizeLimit;

$Apache2::SizeLimit::MAX_PROCESS_SIZE = 100000;

$Apache2::SizeLimit::MAX_UNSHARED_SIZE = 75000;

$Apache2::SizeLimit::CHECK_EVERY_N_REQUESTS = 5;



1;
```

You'll need to install a Perl module to make the above file work. To do that, run the following commands:

```
perl -MCPAN -e shell
install Apache2::SizeLimit
exit
```

Do not yet start or restart Apache, as you still have to install WebGUI and add a site before it will successfully restart.

WebGUI

You need to download the latest version of WebGUI from www.getwebgui.com and extract it to /data/WebGUI. It is possible to install it in another location, but you'll then need to modify the sbin/preload.perl program that comes with WebGUI, and the instructions given here to match your desired location. You should also read the chapter about the /data partition before making this decision.

To do this, run the following commands:

```
mkdir /data
cd /data
tar xvfz /path/to/webgui-7.4.8-stable.tar.gz
```

Now, you need to configure the basic WebGUI system settings. To do this, run the following commands:

```
cd /data/WebGUI/etc
cp log.conf.original log.conf
cp spectre.conf.original spectre.conf
```

If you would like your logs to go to a specific place, then edit the path in log.conf. Other than that, you can leave the default settings, or read the chapter on logging later in this book to make additional modifications.

If you'll be running more than one site on this machine, then you'll want to edit spectre.conf to have a maxWorkers of 5, and a timeBetweenRunningWorkflows of 1. Other than that, you can leave the default settings, or read the chapter on Spectre later in this book to make additional modifications.

Perl Modules

Finally, you'll need to install all of the Perl modules that WebGUI requires. To get a complete list of the Perl modules you'll need, you'll need to run the following commands:

```
cd /data/WebGUI/sbin
perl testEnvironment.pl –simplereport
```

Once you know which Perl modules you need to install, you can install them using the following commands:

```
perl -MCPAN -e shell
install DBI
exit
```

Replace “DBI” with the name of the module you wish to install, like “DBD::mysql” or “Net::SMTP”. There are more than 50 modules that you'll need to install in this way, and some of those have several prerequisite modules that will need to be installed as well. The CPAN system generally makes this easier by automatically downloading and installing prerequisites. However, depending upon your system, this process may or may not go smoothly. This is another reason to use the WRE.

Source Install Complete

You've now completed the source based installation, but your work is not done. You still have to add a site. See the next chapter on how to do that.

Add A Site

As with most things in WebGUI, there's an easy way and a hard way to add a new site for WebGUI. If you used the WRE install, good job. You now have two ways to add a site in under a minute. If you did a source install, good job. That was a lot of work. But, the bad news is it's going to take a while to add a site.

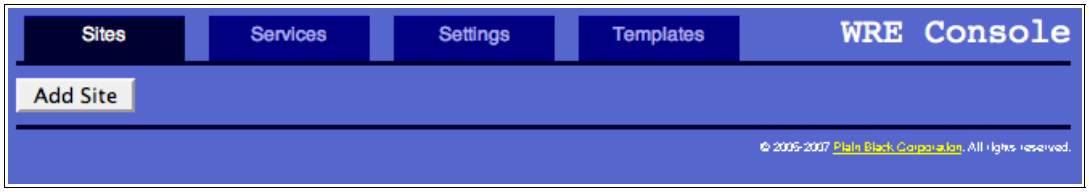
Add A Site With The WRE Console

To add a site with the WRE Console follow the following simple steps.

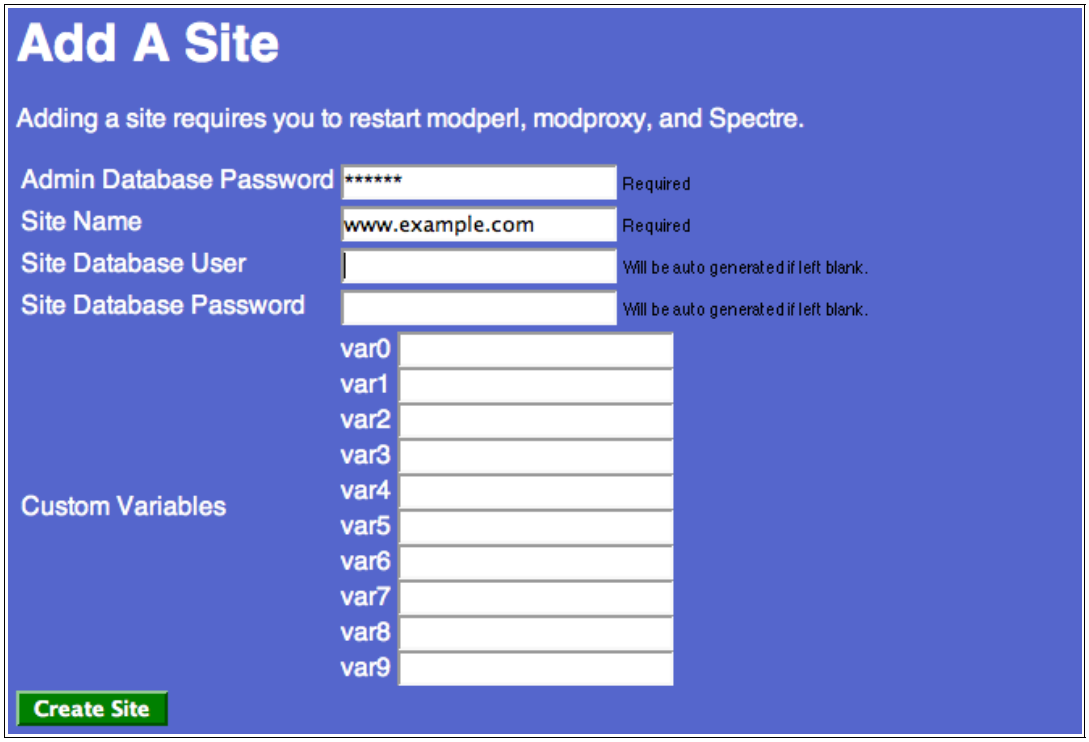
1. If you don't already have the WRE Console running, you can start it with this command:

```
/data/wre/sbin/wreconsole.pl
```

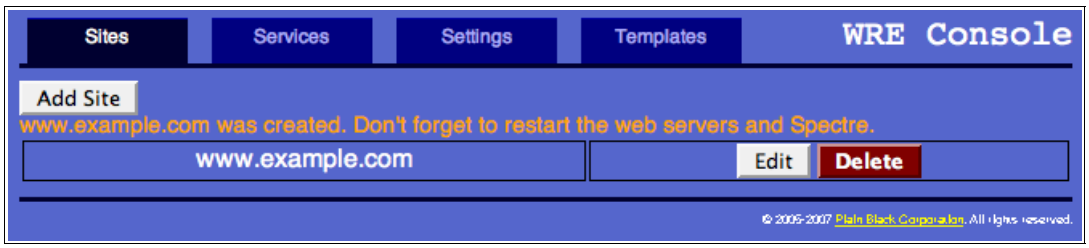
2. Now go to the WRE console with your browser and click on the “Sites” tab if you're not already there.



3. Click on the “Add Site” button. This will bring you to a screen that will let you determine some properties for your new site. All that's really important to fill out is the Admin Database Password (you specified this during the initial configuration of the WRE), and the Site Name (like “www.example.com”).



4. Now, simply click the “Create Site” button and wait about 20 (may be much slower or faster depending upon the speed of your machine) seconds while your new site is configured. If all goes well you'll see a screen that looks like this.



Add A Site With The WRE Command Line

Adding a site from the command line couldn't be simpler. Simply run the following command:

```
/data/wre/sbin/addsite.pl --sitename=www.example.com --adminPassword=123qwe
```

Be sure to replace “www.example.com” with the site name you wish to create, and “123qwe” with your actual database administrator password. If all goes well, you should see output similar to the following:

```
www.example.com was created. Don't forget to restart the web servers and Spectre.
```

Add A Site Manually

Adding a site manually is a multi-step process, but if you follow these steps precisely, you'll have a happy working site at the end.

1. Create a site's web root using the following commands:

```
mkdir -p /data/domains/www.example.com/public
cp -Rf /data/WebGUI/extras/uploads /data/domains/www.example.com/public/
chown -R nobody /data/domains/www.example.com/public/uploads
```

   Be sure to replace “www.example.com” with your site name throughout this process. Also, the chown command uses “nobody” as an example, but it should be whatever user your Apache runs as, which on some servers will be “nobody”, some will use “www”, some will use “httpd”, some will use “apache”, and some will use something else.

2. Now create the site database using these commands:

```
mysql -uroot -p123qwe -e “create database www_example_com”

mysql -uroot -p123qwe -e “grant all privileges on www_example_com.* to  \

        siteuser@localhost identified by 'sitepassword'”

mysql -uroot -p123qwe www_example_com < /data/WebGUI/docs/create.sql
```

   Be sure to replace “siteuser” and “sitepassword” with the username and password you'd like the site to use to access the database. Also note that when creating the databases underscores “_” are used rather than dots “.” between the parts of the domain name. That's because dots have a special meaning in the database world.

3. Create the site config file by running the following commands:

```
cd /data/WebGUI/etc

cp WebGUI.conf.original www.example.com.conf
```

4. Now you need to edit the site config file to match your settings. All the changes you need to make are lumped into one block here, but you will find these config file directives throughout your config file.

```
“sitename” : [ “www.example.com” ],

“uploadsPath” : “/data/domains/www.example.com/public/uploads”,

“dsn” : “DBI:mysql:www_example_com”,

“dbuser” : “siteuser”,

“dbpass” : “sitepassword”,

“spectreSubnets” : [ “127.0.0.1/32”, “10.0.0.0/24”, “192.168.0.1/32” ],
```

5. Be sure to add every IP address, or IP range, attached to the machine where Spectre will run (probably this machine) in the spectreSubnets directive.

6. Finally, you need to create the Apache virtual host that this site will be served from. Edit your httpd.conf file and add the following to the end of it:

```
<VirtualHost *:80>

        ServerName www.example.com

        ServerAlias example.com

        DocumentRoot /data/domains/www.example.com/public

        SetHandler perl-script

        PerlInitHandler WebGUI

        PerlSetVar WebguiConfig www.example.com.conf

</VirtualHost>
```

Of course your virtual host configuration may vary depending upon your system, and what else you have configured in Apache. The above configuration assumes that you have configured the NameVirtualHost directive like this:

```
NameVirtualHost *:80
```

You should now be able to restart Apache and have a working WebGUI site. You can use this procedure again if you want to add an additional site.

Backups

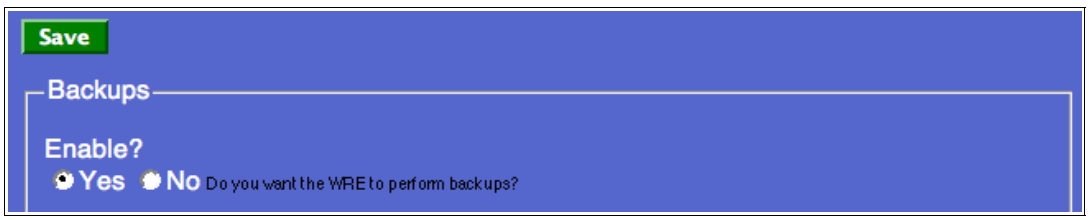
As an administrator, backups should be your best friend. Youshould always make sure you have a good backup at minimum weekly (but every day would be much better), and before every upgrade, just in case something goes wrong. The WRE provides a solid and simple backup system, but just in case you chose to do a source install, we'll show you what you need if you want to manually back up.

WRE

WRE backups are pretty simple to configure and perform. This section shows you how to do it.

1. Start up the WRE Console and go to the Settings tab.

2. Now scroll down to the Backups section and enable backups.



3. Technically you could hit “Save” now and backups would be enabled, but there are lots of other options to configure, so this covers them as well.

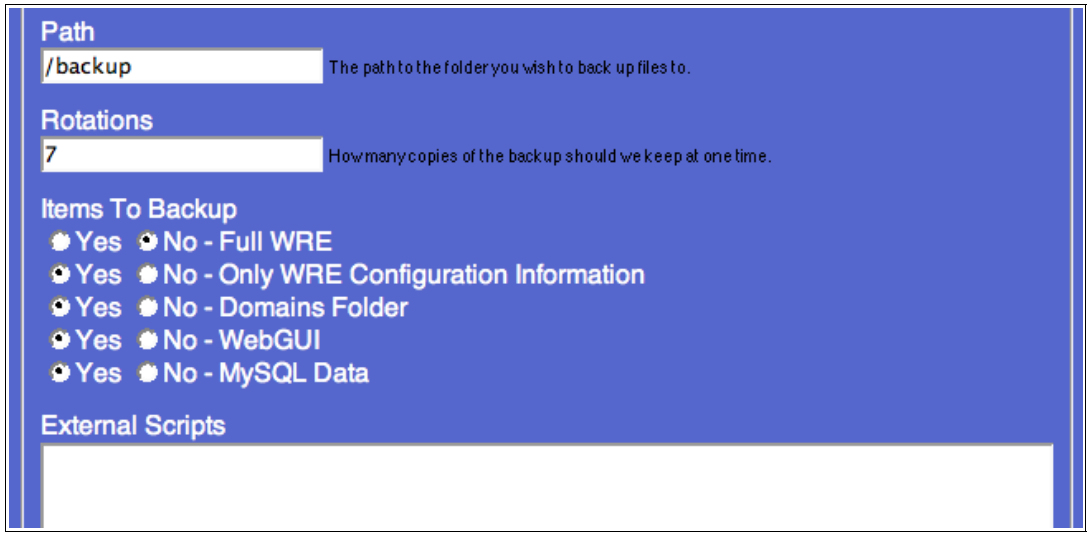
    A. Path tells the backup system where you'd like to store the backup files. Ideally this will be a different drive than the one where your original data is located to protect you in the event of hardware failure.

    B. Rotations is how many copies of the backup data you'd like to keep around. If you're running the backups nightly (which is the default), then this is the number of days worth of data you'll have

on hand. You should probably set this for as high as you have disk space. If your site uses 1GB and you have a 100GB drive to store backups on, then you can reasonably keep 90 to 120 rotations (accounting for compression).

C. Items To Backup allows you to tell the WRE exactly what you want it to back up. The defaults are usually fine, but if you have some other backup system taking care of some portion of this, then you may wish to configure it differently.

D. External Scripts allows you to write your own scripts to plug in to the backup system. You just enter the full path to your script(s) in the box (one per line). If you make sure that the files it creates are put into the backup path, and are GNU Zipped (ends in .gz) then they'll automatically be rotated and passed along to the FTP server if you're using that option.



4. You may also optionally push your backups to an FTP server on another machine. To configure this, fill out the FTP properties.

5. Set Enabled to yes if you wish to use the FTP option.

A. Passive Transfers tells the WRE to enabled the PASSV command while transferring data. This is necessary if your FTP server is behind a firewall or can't open a separate data port for transmission.

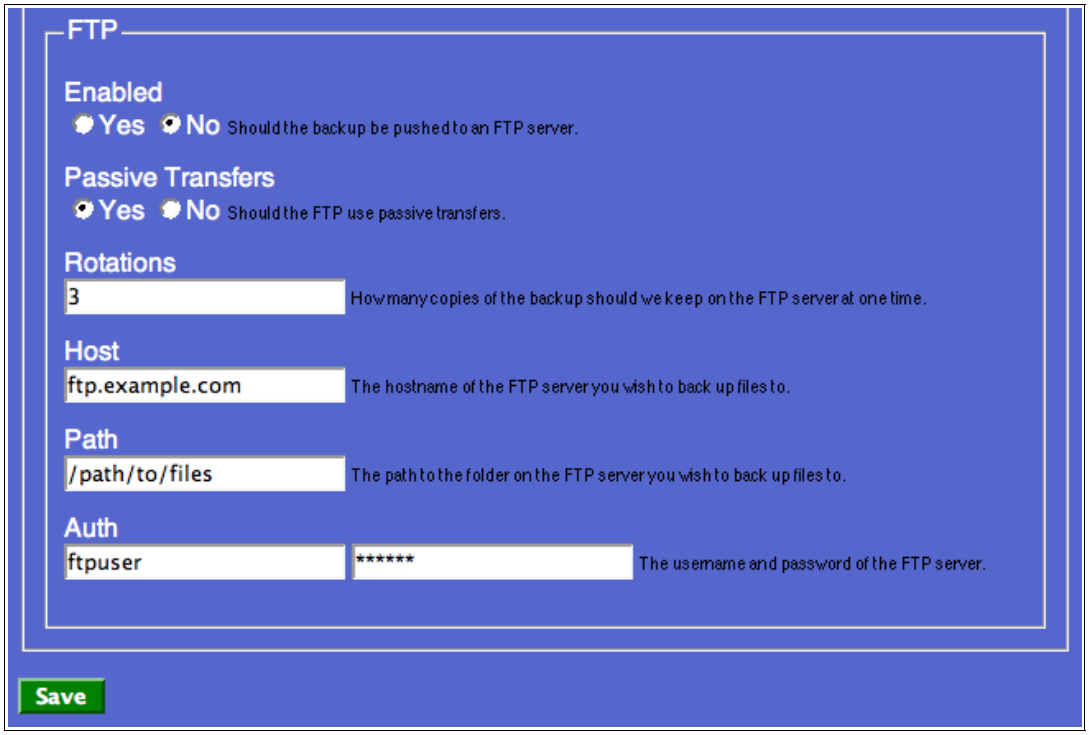
B. Rotations is the same as the backup rotations setting, except

that this is how many rotations will be kept on the FTP server.

C. Host is the name or IP address of the machine where your FTP server lives.

D. Path is the path to the directory on the FTP server where you'd like to upload files. Note that this should be an empty folder on your FTP server, where no other files or folders exist. If it is not those files and folders will get rotated out of existence by the backup process.

E. Auth is the username and password used to connect to the FTP server.



6. Hit “Save” when you are done.

Backups will now be processed every night at 2am (02:00), unless you have changed the default time when you set the cron job during the install process.

Manual

The following process will show you how to back up your WebGUI files manually. Since you're going to want to back up your sites regularly, you'll probably want to turn this into a script that can be run by a cron job.

WebGUI

To back up WebGUI run the following commands:

```
cd /data

tar cvfz /path/to/backup/webgui.tar.gz WebGUI
```

Databases

To back up a site's database run the following commands:

```
mysqldump -uroot -p123qwe www_example_com  > \

        /path/to/backup/www_example.com.sql
cd /path/to/backup

gzip www_example_com.sql
```

Repeat this process once for each site's database.

While it may be tempting to just back up the raw database files, it is not safe to do that unless you shut down MySQL completely; therefore, mysqldump is much safer. In addition, the file produced by mysqldump can be transported across architectures, where the binary database files cannot.

Web Roots

To back up each site's web root run the following commands:

```
cd /data/domains/www.example.com

tar cvfz /path/to/backups/www.example.tar.gz *
```

Repeat these commands once for each site.

Restoration

The restoration process is the same whether you are using WRE backups, or you have manually backed up your data. Follow these steps.

Restore WebGUI by running the following commands:

```
cd /data
rm -Rf WebGUI
tar xvfz /path/to/backup/webgui.tar.gz
```

Restore a site's database by running the following commands:

```
mysql -uroot -p123qwe -e “drop database www_example_com”

mysql -uroot -p123qwe -e “create database www_example_com”

gunzip /path/to/backup/www_example_com.sql.gz

mysql -uroot -p123qwe www_example_com < /path/to/backup/www_example_com.sql
```

Restore a site's web root by running the following commands:

```
rm -Rf /data/domains/www.example.com/*

cd /data/domains/www.example.com

tar xvfz /path/to/backup/www.example.com.tar.gz
```

Upgrades

Upgrading WebGUI is relatively simple and painless, as long as you're prepared. Read this entire chapter before attempting an upgrade.

Upgrade History

If you want to see what upgrades have been applied to your WebGUI install, the upgrade utility can tell you. Simply run this command:

```
cd /data/WebGUI/sbin

perl upgrade.pl --history --doit
```

You'll get output that looks like this, which tells you which upgrades have been applied to each site:

```
Testing site config versions…
Getting site configs...
    Processing www.example.com.conf.

Displaying upgrade history for each site.

www.example.com.conf
    7.3.4   2007-01-17      initial install
    7.3.5   2007-01-17      upgrade
    7.3.6   2007-01-25      upgrade
    7.3.7   2007-01-25      upgrade
```

Note that it tells you which version numbers were applied, the date of the application, and also whether it was the initial installation, or an upgrade.

Upgrade history can be particularly useful when trying to diagnose problems that may have been caused by an old upgrade. For example, if in version 7.0.1 there was something that tweaked a particular setting you're now having trouble with, and you're now running 7.5.5, you can look back and see if the 7.0.1 upgrade was ever applied to your install. Perhaps you started with version 7.5.0, in which case it will be obvious that the 7.0.1 upgrade cannot be responsible for your current problem.

How Upgrades Work

Upgrades in WebGUI work by applying incremental patches to your existing WebGUI site. You first start by copying the new WebGUI distribution over the top of your existing one.

Then, you apply the data patches using the methods described below. This applies each patch to each of your WebGUI site's data since your last upgrade to make your data compliant with the new version of the code. So, if you're upgrading from 7.0.2 to 7.2.3 there might be several dozen patches applied to your site's data. It then repeats this process for every site you have configured. If you have a lot of sites, or you don't upgrade very often, the upgrade process can take a long time, so be prepared for down time.

Note that the upgrade process will overwrite any files you've customized.

Know Your Versions

The first thing to do is to find out if a new version of WebGUI is available. You can do this quickly and easily three different ways:

1. Visit getwebgui.com, which always has the latest version.
2. From your web site go to Admin Console > Statistics and it will tell you what version you are running, and what the latest version is.
3. From your server command line run the following command to find out your local version and the latest:

```
cd /data/WebGUI/sbin

perl testEnvironment.pl
```

This will deliver a lot of output, but what you want is near the end and will look like this (the bold line is the one you're looking for):

```
WebGUI modules:                 OK
Getting current WebGUI version: OK
Your version:                   You are using 7.3.2-stable and 7.3.8-stable is
available.
Locating WebGUI configs:        OK

Checking config file:           www.example.com.conf
Verifying uploads folder:       OK
```

```
Verifying DSN:                  OK
Verifying database connection:  OK

Testing complete!
```

Once you know what version is available, you should also check your own version to see if you're out of date. Methods 2 and 3 (above) will tell you both your current version and the latest version available.

There is one more way to find out your current version number. Edit lib/WebGUI.pm and look near the top of the file, though it may be as much as 20 lines into the file. There will be a $VERSION variable that will tell you what version the code base is. Here's an example of that:

```
our $VERSION = "7.3.9";
our $STATUS = "stable";
```

Backups

Before you ever perform an upgrade, always run a complete backup of your environment. Usually, upgrades will go smoothly, but if they don't, you're going to want a full backup so you can restore anything and everything. For more information about performing backups, see the chapter called "Backups" later in this book.

Note that WebGUI will perform incremental backups of your site database(s) as it upgrades This should not be misconstrued as a full backup. This does not relieve you of your obligation to maintain full backups and conduct one before you upgrade.

Upgrade Options

After you've found out if there is an upgrade, and have decided to upgrade, there are two ways to do it. First, if you're using the WRE you can use WebGUI Update, which is an automated upgrade system. If you did a source install, you can do a manual upgrade.

WRE WebGUI Update

To use WebGUI Update go to the command line, and then do the following:

1. Stop Spectre.

```
/data/wre/sbin/wreservice.pl ---stop spectre
```

2. Run WebGUI Update and follow the onscreen instructions.

```
/data/wre/sbin/webguiupdate
```

3. Restart Apache and start Spectre.

```
/data/wre/sbin/wreservice.pl --restart web --
```

4. Test your site(s) to make sure everything is working.

Whether you do upgrades manually or using WebGUI Update, if the upgrade displays errors and the gotchas don't tell you to expect it, then your upgrade has likely failed. Restore from backup.

Manual Upgrade

To perform a manual upgrade, run the following steps:

1. Download the new version of WebGUI from getwebgui.com.

2. Decompress the new version of WebGUI.

```
cd /tmp
tar xvfz /path/to/webgui-7.5.0-stable.tar.gz
```

3. Read the WebGUI/docs/gotcha.txt file to see what you should know about before upgrading. Follow any instructions in there.

```
more /tmp/WebGUI/docs/gotcha.txt
```

4. Read the WebGUI/docs/changelog/7.x.x.txt file to see what has changed since your current version.

```
more /tmp/WebGUI/docs/changelog/7.x.x.txt
```

5. Stop Spectre.

```
cd /data/WebGUI/sbin

perl spectre.pl --shutdown
```

6. Decompress the new version of WebGUI over the top of the old version of WebGUI

```
cd /data

tar xvfz /path/to/webgui-7.5.0-stable.tar.gz
```

7. Restart Apache.

8. Run the upgrade.

```
cd /data/WebGUI/sbin

perl upgrade.pl
```

Note that the above command won't work directly. You need to read the output of the upgrade script to find out how to make it work. This is our way of ensuring that you're not blindly following instructions without knowing what you're getting into.

9. Run testEnvironment.pl.

```
cd /data/WebGUI/sbin

perl testEnvironment.pl
```

10. Restart Apache again to make it re-read the config files.

11. Start Spectre.

```
cd /data/WebGUI/sbin

perl spectre.pl --daemon
```

12. Test your site(s) to make sure everything is working.

WebGUI Config File

The WebGUI config file provides you with dozens of settings that will enable you to make WebGUI work how you want it to work. The usage of certain portions of the config file will be detailed in other chapters that relate to those specific sections, so if you need information not provided here you can take a look at the other chapters. This chapter is here to describe the uses of each of the available config file directives.

sitename

The sitename field is used throughout WebGUI for generating fully qualified URL's, and also doing hostname validation. This is a list, but generally only the first name in the list is used. For example, the first name in the list is the one Spectre is going to attempt to contact when running a workflow.

```
"sitename" : [ "www.example.com", "example.com"],
```

sslEnabled

This toggle enables SSL related features in WebGUI. Only enable it if you have configured a valid SSL certificate in Apache for this site.

```
"sslEnabled" : 0,
```

cookieName

WebGUI uses a cookie to tie a session to a site visitor. The name of that cookie is traditionally wgSession. However, in some circumstances where you have other custom applications on the same domain, you may want the cookie to be a different name.

```
"cookieName" : "wgSession",
```

cookieDomain

Normally, WebGUI uses the full host and domain name that the site visitor visits the site with to set the cookie. For example, it might use www.example.com. If you have multiple applications on various hostnames you may want to force the cookie to the domain name without the host.

```
"cookieDomain" : ".example.com",
```

cookieTTL

Cookies expire after a certain amount of time. By default, in WebGUI the cookie is set to expire after 10 years. However, in some environments, especially with shared computers, like a school, it's often better to expire the cookie after an hour, or after a day. Here are some examples:

+1h	1 hour
+2D	2 days
+3M	3 months
+10Y	10 years
session	Until browser is closed

```
"cookieTTL" : "+10Y",
```

gateway

Normally, WebGUI is installed to be the only application on a site. However, sometimes you may want to install it as a subdirectory on your site. If that's the case then you'd modify the gateway to match your subdirectory.

```
"gateway" : "/",
```

caseInsensitiveOS

Windows and Mac OS X both use case insensitive file systems. This can cause problems for WebGUI. However, WebGUI can work around those problems if it knows about this problem. Set this if you're using an operating system with a case insensitive file system.

```
"caseInsensitiveOS" : "1",
```

extrasURL

The extras URL tells WebGUI where it should point a user's web browser to find all the images, javascripts, cascading style sheets, and other auxiliary files that WebGUI needs to operate. Normally, this is a relative URL

because it's served from the same machine as WebGUI itself. However, you can provide a full URL if you serve all your static files from another machine, like http://images.example.com/extras

```
"extrasURL" : "/extras",
```

extrasPath

The local filesystem path that WebGUI needs in order to find the extras folder on the local machine.

```
"extrasPath" : "/data/WebGUI/www/extras",
```

uploadsURL

The uploadsURL is the same as extrasURL, except it is for files that users upload to WebGUI.

```
"uploadsURL" : "/uploads",
```

uploadsPath

The uploadsPath is the same as the extrasPath, except that it is for files that users upload to WebGUI.

```
"uploadsPath" : "/data/domains/www.example.com/public/uploads",
```

richEditorsUseAssetUrls

This option tells WebGUI's rich editors to use the short human friendly asset URL's when linking to images and files rather than the URL to the uploads folder. The advantage to this is that the URL's are prettier, and they continue to work even if a user edits the image. However, it's slower than the default.

```
"richEditorsUseAssetUrls" : 0,
```

passthruUrls

If you want to put WebGUI at the root of your site, but you want to serve up other applications and files outside of WebGUI, you can put those paths here.

```
"passthruUrls" : ["/icons", "/documentation/pdf", "/my-custom-application", "/server-
```

```
status", "/perl-status"],
```

For each item in your passthruUrls directive, you'll also need to tell Apache that you want it to pass through, so configure a block like this in your Apache config:

```
<Location /documentation/pdf>
        SetHandler None
</Location>
```

webServerPort

This directive tells both WebGUI and Spectre what port to generate URL's for when connecting to the server.

```
"webServerPort" : 80,
```

cacheType

The cacheType directive tells WebGUI which caching engine to use to cache data for a performance boost. In most situations you should use WebGUI::Cache::Database. But if you're running small sites (less than a few hundred pages) WebGUI::Cache::FileCache will be faster.

```
"cacheType" : "WebGUI::Cache::FileCache",
```

fileCacheRoot

If you are using WebGUI::Cache::FileCache, you can specify where to write the cache using this directive.

```
"fileCacheRoot" : "/path/to/cache",
```

disableCache

If you use disableCache then all caching subsystems will be disabled in WebGUI. This is sometimes useful for troubleshooting, but it will severely impact the performance of your server.

```
"disableCache" : 0,
```

dsn

The dsn directive configures the database driver for WebGUI to use. You

can tell it what database name to use, the hostname of the MySQL server, and even the port of the MySQL server.

```
"dsn" : "DBI:mysql:www_example_com;host=localhost",
```

dbuser

This is the username that will be used to connect to the MySQL database.

```
"dbuser" : "webgui",
```

dbpass

This is the password that will be used to connect to the MySQL database.

```
"dbpass" : "password",
```

dbSlave1, dbSlave2, dbSlave3

If you have database slaves, you can use them to slightly improve performance. You can configure up to three slaves using these directives.

```
"dbslave1" : {
   "dsn" : "DBI:mysql:www_example_com;host=dbslave1.example.com",
   "user" : "webgui",
   "pass" : "password"
},
```

failoverdb

The failoverdb directive says that if the master database cannot be connected to, then it should try to connect to this database.

If you are in a type of environment that absolutely must never go down, not even for a single request, then you can use this option. However, there are usually better ways to accomplish what this function does. The biggest problem with this function is that if the master database only goes down for a little while, and then comes back up, there is no way to transfer the data that was written to the failover database back to the master in an automated fashion. You'll need to write scripts for this purpose. Or better yet, use a more robust failover mechanism.

```
"failoverdb" : {
    "dsn" "DBI:mysql:www_example_com;host=failover.example.com;port=3306",
    "user" : "webgui",
```

```
    "password" : "password"
},
```

emailOverride

If you're testing an email function, and don't wish to panic your users until after you've tested it, then use this directive to redirect all outgoing email to this address.

```
"emailOverride" : "joe@example.com",
```

adminModeSubnets

If you want to require that no one can go into admin mode on your site unless they are within your network, then you can add a list of subnets in CIDR notation into this directive:

```
"adminModeSubnets" : [“10.0.1.0/24”,”198.162.0.1/32”],
```

authMethods

WebGUI allows for pluggable authentication modules. Specify the ones you wish to be used on your site here.

```
"authMethods" : [ "LDAP", "WebGUI" ],
```

paymentPlugins

WebGUI allows for multiple payment gateways. Use this directive to enable the ones you want to use.

```
"paymentPlugins" : ["Itransact","Cash"],
```

shippingPlugins

The commerce system also allows for multiple shipping gateways. Use this directive to enable those.

```
"shippingPlugins" : ["ByPrice", "ByWeight", "PerTransaction"],
```

templateParsers

By default WebGUI uses WebGUI::Asset::Template::HTMLTemplate as its template parser. However, you can also use HTMLTemplateExpr and

TemplateToolkit for more advanced templates,or you could write your own.

```
"templateParsers" : ["WebGUI::Asset::Template::HTMLTemplate"],
```

defaultTemplateParser

In the event that no template parser is selected on a template, this directive will use the template parser you specify.

```
"defaultTemplateParser" : "WebGUt:Asset::Template::HTMLTemplate",
```

searchIndexerPlugins

You can specify external programs to help index content in uploaded files. There is more on this directive in the “Search Indexer” chapter.

```
"searchIndexerPlugins" : {
   "txt" : "/bin/cat",
   "readme" : "/bin/cat",
   "html" : "/bin/cat",
   "htm" : "/bin/cat"
   },
```

maximumAssets

If you want to allow a maximum number of assets that a user can have on their site in order to force their site to remain small, then you can set this directive. Keep in mind that there are around 300 assets that come with WebGUI. If you set it to zero then the site may have an unlimited number of assets.

```
"maximumAssets" : 0,
```

assets

This directive is a list of asset class names that are available to users to add to the site.

```
"assets" : [
   "WebGUI::Asset::Snippet",
   "WebGUI::Asset::Redirect",
   "WebGUI::Asset::FilePile",
   "WebGUI::Asset::Wobject::Article",
   "WebGUI::Asset::Wobject::Collaboration",
   "WebGUI::Asset::Wobject::DataForm",
   "WebGUI::Asset::Wobject::Calendar",
```

```
  "WebGUI::Asset::Wobject::EventManagementSystem",
  "WebGUI::Asset::Wobject::HttpProxy",
  "WebGUI::Asset::Wobject::Navigation",
  "WebGUI::Asset::Wobject::Matrix",
  "WebGUI::Asset::Wobject::Poll",
  "WebGUI::Asset::Wobject::Product",
  "WebGUI::Asset::Wobject::ProjectManager",
  "WebGUI::Asset::Wobject::SQLReport",
  "WebGUI::Asset::Wobject::Search",
  "WebGUI::Asset::Wobject::Survey",
  "WebGUI::Asset::Wobject::TimeTracking",
  "WebGUI::Asset::Wobject::WeatherData",
  "WebGUI::Asset::Wobject::MultiSearch",
  "WebGUI::Asset::Wobject::StockData",
  "WebGUI::Asset::Wobject::SyndicatedContent",
  "WebGUI::Asset::Wobject::InOutBoard",
  "WebGUI::Asset::File::ZipArchive",
  "WebGUI::Asset::Wobject::WSClient",
  "WebGUI::Asset::Wobject::SQLForm"
  ],
```

utilityAssets

This is the same as the assets directive, except that these assets are rarely used and therefore are only displayed in the asset manager view.

```
"utilityAssets" : ["WebGUI::Asset::Template", "WebGUI::Asset::RichEdit",
"WebGUI::Asset::File::Image", "WebGUI::Asset::File"],
```

assetContainers

These assets act as containers for other assets. These assets are typically layout mechanisms rather than content.

```
"assetContainers" : [
  "WebGUI::Asset::Wobject::Layout",
  "WebGUI::Asset::Wobject::Folder",
  "WebGUI::Asset::Wobject::Dashboard",
  "WebGUI::Asset::Wobject::MessageBoard",
  "WebGUI::Asset::Wobject::WikiMaster"

  ],
```

assetAddPrivilege

The assetAddPrivilege directive allows you to define what group a user must be in to add given types of assets.

```
"assetAddPrivilege" : {
  "WebGUI::Asset::Wobject::SQLReport" : 3,
  "WebGUI:::Asset::Template" : 4
  },
```

enableSaveAndCommit

If this directive is enabled then WebGUI will show two buttons instead of the normal one save button on asset editing pages. The first is the normal save button, but the second is the “Save and Commit” button. The second button saves the current asset, and commits the version tag at the same time.

```
"enableSaveAndCommit" : 0,
```

assetUiLevel

This directive allows you to set the UI Level that the user must have in order to see a given asset type in the add content menu.

```
"assetUiLevel" : {
  "WebGUI::Asset::Wobject::WSClient" : 7,
  "WebGUI::Asset::RichEdit" : 4
  },
```

assetToolbarUiLevel

Assets have a lot of common functions in their toolbars. This directive allows you to control what UI level your users must have to see each of the toolbar options.

```
"assetToolbarUiLevel" : {
   "edit" : 1,
   "delete" : 1,
   "cut" : 1,
   "copy" : 1,
   "shortcut" : 5,
   "editBranch" : 9,
   "lock" : 5,
   "export" : 9,
   "changeUrl" : 9,
   "promote" : 3,
   "demote" : 3,
   "manage" : 5,
   "revisions" : 5,
   "view" : 1
   },
```

Asset Field UI Levels

It takes quite a bit of work, but you can specify the UI levels of every field in every asset. In this way you can truly customize the user experience. Here you can see examples of how to specify the UI levels for a couple of fields in two different assets.

```
"WebGUI_Asset_Wobject_Article_uiLevel" : { "menuTitle" : 9, "url" : 8 },

"WebGUI_Asset_RichEdit_uiLevel" : { "askAboutRichEdit" : 7, "preformatted" : 3 },
```

exportPath

If you wish to export content to static files, then you need to specify the exportPath directive. The value is the location on the filesystem where the static files will be stored as they are exported.

```
"exportPath" : "/path/to/export",
```

soapHttpHeaderOverride

If you are using the WS Client asset and wish to allow them to override the default mime types in SOAP/WSDL documents, then use this directive.

```
"soapHttpHeaderOverride" : 0,
```

enableStreamingUploads

Normally, if a request comes into an asset URL and that URL represents a file or an image, WebGUI will redirect the browser to the actual file URL. This is done because the redirect, and subsequently letting Apache serve the file directly, has far less impact on performance. However, in some circumstances, mostly for vanity sake, you may want WebGUI to actually serve the file itself. If you do, then enable this option. Be warned that this option adversely impacts performance.

```
"enableStreamingUploads" : "0",
```

macros

This directive is a list of name/value pairs where the name is an alias that you want to be processed as a macro on your site, and the value is the filename of the macro code to execute. You are allowed to alias a single macro multiple times if your users are used to a particular name, or

frequently like to use different names.

```
"macros" : {
      "#" : "Hash_userId",
      "/" : "Slash_gatewayUrl",
      "@" : "At_username",
      "AOIHits" : "AOIHits",
      "AOIRank" : "AOIRank",
      "AdminBar" : "AdminBar",
      "AdminText" : "AdminText",
      "AdminToggle" : "AdminToggle",
      "AdSpace" : "AdSpace",
      "AssetProxy" : "AssetProxy",
      "CanEditText" : "CanEditText",
      "D" : "D_date",
      "EditableToggle" : "EditableToggle",
      "Extras" : "Extras",
      "FetchMimeType" : "FetchMimeType",
      "FileUrl" : "FileUrl",
      "GroupAdd" : "GroupAdd",
      "GroupDelete" : "GroupDelete",
      "GroupText" : "GroupText",
      "H" : "H_homeLink",
      "International" : "International",
      "L" : "L_loginBox",
      "LastModified" : "LastModified",
      "LoginToggle" : "LoginToggle",
      "Page" : "Page",
      "PageTitle" : "PageTitle",
      "PageUrl" : "PageUrl",
      "RandomAssetProxy" : "RandomAssetProxy",
      "RandomThread" : "RandomThread",
      "RootTitle" : "RootTitle",
      "Spacer" : "Spacer",
      "SubscriptionItem" : "SubscriptionItem",
      "SubscriptionItemPurchaseUrl" : "SubscriptionItemPurchaseUrl",
      "Thumbnail" : "Thumbnail",
      "User" : "User",
      "a" : "a_account",
      "c" : "c_companyName",
      "e" : "e_companyEmail",
      "r" : "r_printable",
      "u" : "u_companyUrl"
   },
```

ldapAlias

Using the ldapAlias directive you can specify field matches between WebGUI profile fields and LDAP attributes. This is useful when combined with the LDAP synching workflow activities.

```
"ldapAlias" : {
        "firstName" : "givenName",
        "lastName" : "sn",
        "email" : "mail",
        "companyName" : "o"
    },
```

spectreSubnets

This directive is a list of subnets specified in CIDR notation that WebGUI should expect Spectre to connect from. Any connections from subnets other than these will be rejected by WebGUI.

```
"spectreSubnets" : [ "127.0.0.1/32" ],
```

spectreIp

When WebGUI connects to Spectre what IP address should it use?

```
"spectreIp" : "127.0.0.1",
```

spectrePort

This directive goes along with spectreIp to tell WebGUI how to connect to Spectre.

```
"spectrePort" : 32133,
```

workflowActivities

The workflowActivities directive allows you to configure which workflow activities are available to which object types for your users to build new workflows.

```
"workflowActivities" : {
        "None" : [
                "WebGUI::Workflow::Activity::DecayKarma",
                "WebGUI::Workflow::Activity::TrashClipboard",
                "WebGUI::Workflow::Activity::CleanTempStorage",
                "WebGUI::Workflow::Activity::CleanFileCache",
                "WebGUI::Workflow::Activity::CleanLoginHistory",
                "WebGUI::Workflow::Activity::ArchiveOldThreads",
                "WebGUI::Workflow::Activity::TrashExpiredEvents",
                "WebGUI::Workflow::Activity::CreateCronJob",
                "WebGUI::Workflow::Activity::DeleteExpiredSessions",
                "WebGUI::Workflow::Activity::ExpireGroupings",
```

```
            "WebGUI::Workflow::Activity::PurgeOldAssetRevisions",
            "WebGUI::Workflow::Activity::ExpireSubscriptionCodes",
            "WebGUI::Workflow::Activity::PurgeOldTrash",
            "WebGUI::Workflow::Activity::GetSyndicatedContent",
            "WebGUI::Workflow::Activity::ProcessRecurringPayments",
            "WebGUI::Workflow::Activity::SyncProfilesToLdap",
            "WebGUI::Workflow::Activity::SummarizePassiveProfileLog",
            "WebGUI::Workflow::Activity::SendQueuedMailMessages",
            "WebGUI::Workflow::Activity::CleanDatabaseCache",
            "WebGUI::Workflow::Activity::CalendarUpdateFeeds",
            "WebGUI::Workflow::Activity::NotifyAdminsWithOpenVersionTags"
        ],
       "WebGUI::User" : [
            "WebGUI::Workflow::Activity::CreateCronJob",
            "WebGUI::Workflow::Activity::NotifyAboutUser"
        ],
        "WebGUI::VersionTag" : [
            "WebGUI::Workflow::Activity::CommitVersionTag",
            "WebGUI::Workflow::Activity::RollbackVersionTag",
            "WebGUI::Workflow::Activity::TrashVersionTag",
            "WebGUI::Workflow::Activity::CreateCronJob",
            "WebGUI::Workflow::Activity::UnlockVersionTag",
            "WebGUI::Workflow::Activity::NotifyAboutVersionTag",
            "WebGUI::Workflow::Activity::RequestApprovalForVersionTag",
            "WebGUI::Workflow::Activity::ExportVersionTagToHtml"
        ]
       },
```

graphingPlugins

WebGUI has a built in graphing system that can be extended with new plugins. Currently, it is only used in the Poll, but it may also be used in custom applications. Here you can specify a list of plugins that can be used to generate graphs.

```
"graphingPlugins" : [
   "WebGUI::Image::Graph::Pie",
   "WebGUI::Image::Graph::XYGraph::Bar",
   "WebGUI::Image::Graph::XYGraph::Line"
   ],
```

availableDictionaries

If you install the C program aspell and the Text::Aspell Perl module, you can enable WebGUI's built in spell checking system. You simply need to use this directive to tell it which dictionaries are available on your system.

```
"availableDictionaries" : [
   {
```

```
    "id" : "en_US",
    "name" : "English",
    "default" : "1"
  },
  {
    "id" : "nl",
    "name" : "Dutch"
  }
],
```

runOnLogin

This specifies a script to be executed as the user logs in to WebGUI. You can use macros in the command and they will be processed.

```
"runOnLogin" : "/path/to/login.pl --user=^@;",
```

runOnLogout

This specifies a script to be executed as a user logs out of WebGUI. You can use specify macros in the command and they will be processed.

```
"runOnLogout" : "/path/to/logout.pl --user=^@;"
```

Spectre Config

The spectre.conf is used to tell Spectre how to communicate with WebGUI, and also how Spectre should perform its functions.

ip

This directive tells Spectre what IP to bind to.

```
"ip" : "127.0.0.1",
```

port

This directive tells Spectre what port to bind to. This, together with the IP, will be what WebGUI will use to communicate with Spectre.

```
"port" : 32133,
```

maxWorkers

The maxWorkers directive tells Spectre how many workflows it may attempt to run simultaneously. Most workflow activities run very quickly, so Spectre may end up running them only one at a time, but if you have some long running workflow activities then you may reach this limit. You want to configure this high enough that Spectre can get all of its work done in a reasonable amount of time, but not so high that it overwhelms WebGUI. A setting between 3 and 7 is recommended.

```
"maxWorkers" : 3,
```

timeBetweenRunningWorkflows

This directive determines how long (in seconds) Spectre should wait between starting workflow activities. This should normally be 1 or 2 seconds. A setting of 0 could end up with a denial of service to WebGUI depending upon how many workers you have configured, and a setting of much more than two may mean that Spectre gets backlogged with tasks if the server is busy.

```
"timeBetweenRunningWorkflows" : 2,
```

suspensionDelay

When a workflow activity returns “error” or “waiting”, Spectre suspends that workflow for a time, waiting for the error to clear itself up, or for the whatever the workflow is waiting on to happen. This directive determines how long those workflows should be suspended. Generally speaking this should be no lower than 60 and no higher than 300.

```
"suspensionDelay" : 60,
```

webguiPort

This directive tells Spectre what port to connect to WebGUI on.

```
"webguiPort" : 80,
```

ignoreEnvProxy

If your server uses a proxy to gain access to the Internet, but WebGUI is located on your local network, then you should set this to 1 so that Spectre doesn't attempt to use the proxy to connect to WebGUI.

```
"ignoreEnvProxy" : 0
```

Performance Tuning

Performance tuning can mean the difference between having to upgrade your hardware and not. It can mean the difference between having happy users and not. It should be taken every bit as seriously as backups. And just like backups, you should test your environment regularly to make sure that what you did before is still working, or maybe some more tuning is necessary.

The following items are provided in no particular order. Depending upon your particular needs, one may produce more results than the other. There are, unfortunately, no general rules as to which of these is best suited to your environment, budget, time constraints, or needs. The best advice is to consider each one carefully and apply as needed.

MySQL Tuning

There are literally dozens of settings that you could tweak in MySQL. To learn about how each one affects performance check out the MySQL manual at www.mysql.com. However, there are a few we can shed some light on right here.

Edit your my.cnf file.

```
vi /data/wre/etc/my.cnf
```

Adjust each of the following relative to each other at an appropriate level for the amount of RAM you have available on this machine. If you're running MySQL on the same machine as the rest of WebGUI, you have to take that into account. But if you're running MySQL on its own machine, and it has 4GB of RAM free, then you could easily quadruple these numbers:

```
key_buffer=256M
myisam_sort_buffer_size=64M

sort_buffer=2M

record_buffer=2M

thread_stack=128K

query_cache_size=32M

table_cache=512
```

The thread_concurrency setting should be set to double the number of processors you have available.

```
thread_concurrency=4
```

The thread_cache setting should be set high enough that when you run “show status” from the MySQL command line the threads_created value increases very little over time while your server is under load.

```
thread_concurrency=12
```

As far as WebGUI is concerned, the ideals for the following settings are as follows:

```
max_connections=1000

wait_timeout=600
```

After you've made these changes you'll need to restart MySQL for them to take effect.

MySQL Slaves and Replication

By using MySQL replication you can increase performance in two ways. The first is that you can add the MySQL slave to your WebGUI config file and it will be used for low priority read requests, which will somewhat reduce the load on the master server. In addition, you can use mysqldump to back up against the slave rather than the master, which means that your backups won't have any adverse performance effects on your site.

There is an additional benefit to using a slave. The slave gives you an always on backup of your database. Therefore, if your database server ever goes down you can quickly switch over to use the slave as the master.

The good folks at MySQL have done an excellent job describing how to use replication. You can set up MySQL replication using the replication how-to: http://dev.mysql.com/doc/refman/5.0/en/replication-howto.html

The basic steps are as follows:

1. Start with:

```
vi /data/wre/etc/my.cnf
```

2. Add the following line to the config:

```
log-bin=mysql-binserver-id=1
```

3. Restart MySQL.

```
/data/wre/sbin/wreservice.pl --restart mysql
```

4. Log in to the server and perform the following commands:

```
mysql -uroot -p123qwe

grant replication slave on *.* to 'repl'@'10.0.0.2'  identified by 'somepassword';

flush privileges;

flush tables with read lock;

show master status;
```

```
+---------------+----------+--------------+------------------+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+---------------+----------+--------------+------------------+
| mysql-bin.001 | 92       |              |                  |
+---------------+----------+--------------+------------------+
```

5. Copy down the resulting information and keep it safe.

6. Using another terminal **(do not quit your mysql client session)** log in to the server again, and make a tarball of all the databases you want to replicate.

```
cd /data/wre/var/mysqldata

tar cvfz /tmp/mysqlsnapshot.tar.gz www_example_com mysql
```

7. In your mysql client, you can safely unlock the tables and exit:

```
unlock tables;

exit
```

8. Copy your snapshot over to your slave server. Stop the slave. Extract your snapshot and edit MySQL config on the slave server:

```
rm -Rf /data/wre/var/mysqldata/*
cd /data/wre/var/mysqldata

tar xvfz /tmp/mysqlsnapshot.tar.gz

vi /data/wre/etc/my.cnf
```

9. And add the following line to the slave:

```
server-id=2
```

10. Start the slave. Log in and run the following commands:

```
mysql -uroot -p123qwe

change master to master_host='10.0.0.1', master_user='repl',
master_password='somepassword', master_log_file='mysql-bin.001', master_log_pos=92;
start slave;
```

You now have basic MySQL Master/Slave replication set up.

mod_ perl Tuning

There are many things you can do to tune mod_perl to work better and faster. In the WRE, all of these things have been preset to be as good as they can be without knowing about your environment and traffic patterns.

On platforms that support Apache2::SizeLimit (which is everything except Mac OS X, as far as we know) you should edit modperl.pl (in /data/wre/etc/ or /etc/) to match your needs.

MAX_PROCESS_SIZE is the largest size that mod_perl will allow its processes to grow before killing them. On systems where you have pruned out a bunch of WebGUI plugins, this can go lower. On systems where you have a bunch of your own custom plugins, or if you have a lot of extra RAM, this can go higher.

MAX_UNSHARED_SIZE is similar to MAX_PROCESS_SIZE except that it kills the process if the amount of RAM not shared with the master Apache process is greater than this value.

CHECK_EVERY_N_REQUESTS should be set low enough to catch processes that are growing very large, but high enough that the check doesn't stress the server. A setting of 5 is usually about right.

```
$Apache2::SizeLimit::MAX_PROCESS_SIZE = 100000;

$Apache2::SizeLimit::MAX_UNSHARED_SIZE = 75000;

$Apache2::SizeLimit::CHECK_EVERY_N_REQUESTS = 5;
```

If you are not able to use Apache2::SizeLimit, then you can set MaxRequestsPerChild in your /data/wre/etc/modperl.conf or your

httpd.conf if you're running a source install. This should be set to something 1000 or less depending on how quickly your processes are growing. You also need to take into account how busy your server is. The lower this number the more it will stress the server because it will be creating new processes more frequently.

```
MaxRequestsPerChild 1000
```

Also in your /data/wre/etc/modperl.conf or httpd.conf you should turn off keep alives. Keep alives are good for serving static files, but bad for memory hungry mod_perl processes.

```
KeepAlive Off
```

And finally, in your /data/wre/etc/modperl.conf or httpd.conf, you should set the process directives according to your traffic patterns and available memory.

StartServers and MinSpareServers should both be set to the average number of mod_perl processes that will be serving requests at any given time.

MaxSpareServers should be set to the maximum number of idle servers you want left around in case of a load spike.

MaxClients should be set to the maximum number of requests you server can simultaneously handle. For example, if you have a gigabyte of RAM available for mod_perl, and your processes max out at 100MB, then MaxClients should be set to 10.

```
StartServers 5

MinSpareServers 5

MaxSpareServers 10

MaxClients 20
```

Don't forget to restart mod_perl after making these changes.

Reverse Proxy Web Server

A reverse proxy is very good for performance in that it can handle serving static files quickly, and then hand off requests to the mod_perl server only when needed. Since the process size of the reverse proxy is 5MB or less per process, you can have a lot of these processes available for serving

static requests.

In addition, the reverse proxy helps take some additional load off the mod_perl server in that slow processing clients (like people on dial-up connections) can be handled by the reverse proxy. mod_perl quickly does its work, hands off the content to mod_proxy, and that mod_perl process is freed up to do some more work while mod_proxy slowly spoon feeds the content back to the slow client. Finally, mod_proxy can offload processor intensive, but low memory functions like SSL and compression.

If you're using the WRE then you already have a reverse proxy server in the form of mod_proxy.

If not, here are some recommendations for reverse proxy servers available:

- Apache mod_proxy - http://httpd.apache.org/docs/2.0/mod/mod_proxy.html
- perlbal - http://www.danga.com/perlbal/
- Litespeed - http://litespeedtech.com/

Load Balancing

Load balancing is the act of distributing requests between two or more servers. This is done via a load balancing device (sometimes called an IP Sprayer) or a piece of software on another server. The performance advantage of load balancing is that you can distribute the requests coming in to your web site to multiple machines. There is an added benefit in that if one machine goes down, the others can continue serving requests, and your users won't notice any down time.

There are many different kinds of load balancing solutions out there. The following instructions show you how to set up one off the smallest and easiest on a standard linux server.

1. Get balance and install it on the the server you wish to act as your load balancer: http://www.inlab.de/balance.html
2. Using it is really simple, just type these commands to get it started:

```
balance http 10.0.0.10 10.0.0.11

balance https 10.0.0.10 10.0.0.11
```

That will then load balance all traffic coming in for http and https between two servers with IP addresses of 10.0.0.10 and 10.0.0.11.

Note that you'll likely want to configure some sort of startup script for balance so that when your machine is rebooted, balance will start automatically.

Network

The way your network is configured can have an enormous impact on the performance of your web site, especially if you have multiple machines working together to serve up your site.

No matter what your web server looks like, it should be on a switched network segment. A lot of new administrators and old networks use ordinary network hubs to connect the machines. If you have 10 machines on a 100 megabit network connected via a hub, then all 10 machines have to share that 100 megabits. But a switch has a wide bandwidth backplane, and routes packets only to the network segments they need to be delivered over. This means that in the same network configuration you might be able to get 500 megabits or more shared between your machines.

If you have multiple machines working together to serve up your site, then they should each have one network interface card (NIC) per direction they are communicating. For example, if you have a web server and a database server, the web server should have one NIC for incoming requests from the Internet, and another NIC to communicate with the database. But since the database has only to communicate with the web server, then it only needs one NIC. However, if you add a slave to that, then both the master and slave will need two NIC's. One for communicating with each other, and one for communicating with the web server. In this way they aren't sharing bandwidth on each NIC for communication with various sources. And of course, all of these NIC's should be connected together on a switched network segment.

Believe it or not, the physical length of the network cable can add latency to your network communication. In addition, each device (hub, router, switch, firewall, bridge) that the packets pass through when traveling over the network can add small amounts of latency. For general purposes (file serving) this amount of latency doesn't add up to much. However, when you're talking about database to web server communication this little bit can make a big difference. For this reason, we recommend that your database and web servers are on the same network segment in the same physical rack. This keeps the cable short and down to only one device that

the traffic must pass through.

This last bit of advice should go without saying, but it is often missed or overlooked, so we'll say it. Look at your networking equipment to see if there is any packet loss, routing errors, or network collisions. Sometimes one bad piece of hardware or a poor configuration can destroy a good network. Incidentally, a lot of network errors are caused in environments that are mixed 10MB and 100MB. Auto sensing ports aren't as good as they should be, so force them to a particular speed. This doesn't happen as much anymore since 10MB networks are mostly phased out, but as people migrate to gigabit networks it is still something to think about.

Hard Disks

Hard disks are very complicated machines these days. In addition, they're also the slowest part of any server, and the most prone to failure, so they need a lot of special attention.

If you notice that the load on your server is getting high, but you aren't serving that many requests, it may be that your hard drives are the bottleneck. It could be that some process (backups, auxiliary functions, serving lots of large files or even more small files) is using all the speed your hard drives have to give. This is sometimes called running up against spindle speed. If this is the case, disabling some processes may fix it, moving to a multi-disk RAID system may fix it, or you might need an additional machine and a load balancer.

RAID (redundant array of inexpensive disks) can be both a lifesaver and a bottleneck. RAID allows you to mirror your data across multiple disks. While this is good for data redundancy, it can also be bad for performance, because every write has to be done at least twice. Thankfully, most people use RAID 0,1 or RAID 5, which has a striping component. This means that your data is spread out (or striped) across multiple disks, which in turn speeds up reads significantly. So what you lose in write speed you gain back, and then some in read speed.

Whether you're using RAID or not, the processing power of your disk controller, coupled with the amount of Cache or RAM it has available, can significantly impact performance. More expensive disk controllers, and especially RAID controllers, come with diagnostic tools that can tell you if you're maxing out the abilities of your controller.

Your disk controller isn't the only thing with cache. The disks themselves have cache built in, as well as an on-disk controller card. These things can

both affect performance. When you're buying your disks make sure you're getting the highest quality disks you can afford.

Spindle speed is another thing that can affect performance. Sometimes manufacturers (and system administrators) will try to cheap out and use desktop drives in a server. Desktop drives spin at between 4800 and 7200 RPM's. However, server drives often spin at speeds of up to 15,000 RPM's. That's more than double the speed of the desktop counterparts.

Finally, there are two main types of hard drives in the world. The IDE family, which includes the SATA line of drives, and the SCSI family. Though there is much debate about whether the super fast SATA II drives can outperform SCSI, it has been Plain Black's experience that SCSI still wins in server environments.

Memory

Often adding some memory to a server can save you from buying a whole new server. This is especially true with memory intensive applications such as WebGUI and MySQL.

A big thing to check on is whether your server is using a lot of virtual memory. This is called “swapping”, as in the machine is swapping real memory for virtual memory on the hard drive. Virtual memory is several orders of magnitude slower than physical memory. You cannot afford to have your server using your hard disk as memory. Therefore if you notice it using virtual memory, get more RAM.

Auxiliary Server Functions

Running extra services on your server can have an adverse impact on your web site's performance. For example, if you're running an FTP server on the box, even though that is typically low in memory and processor usage, it is usually quite high in disk I/O and network I/O. Thus, it's eating up valuable resources that your web site needs. The same is true for other common services. DNS and IRC are heavy on network I/O. Backup servers, log monitors, and web stats processing are heavy on disk I/O. Mail servers, especially with spam filtering, are heavy on everything.

In addition, servers often ship with processes running that you'll never use. No matter how small the amount of resources it's using, it's still using resources. For example, Red Hat Enterprise Linux often ships with gpm and atd enabled. gpm is used to give you a command line mouse pointer,

and atd is an old alternative to cron that basically no one uses anymore. Do you use either of these things? No? Then turn them off.

In general you should migrate any auxiliary processes you need off to some smaller hardware somewhere, and then disable anything you don't need.

WebGUI Modules

WebGUI is a huge system. The fact of the matter is that most people don't use even half of what WebGUI offers. However, by default, all that extra stuff is getting loaded into memory anyway. By eliminating the components you don't need, you might save as much as 10MB per mod_perl process. If you have 20 processes on the machine, that's a 200MB savings, which is no small potatoes.

To eliminate WebGUI components you need to do two things. The first is to eliminate them from all your site's WebGUI config files. The second is to tell mod_perl not to load them into memory.

In your WebGUI config files there are a number of directives to look at. They are:

- authMethods
- paymentPlugins
- shippingPlugins
- templateParsers
- assets
- utilityAssets
- assetContainers
- macros
- workflowActivities
- graphingPlugins

Each of these directives is a list of plugins that are available for use on your site. By eliminating the ones you don't use, you can save a lot of memory. For example, in authMethods, unless you're a business running an intranet,

you probably don't use LDAP, so you might as well removeit. If you're not running an online store you can empty out paymentPlugins and shippingPlugins. The following is an example of the default assets list followed by a probable use assets list for most people:

Before

```
"assets" : [
   "WebGUI::Asset::Snippet",
   "WebGUI::Asset::Redirect",
   "WebGUI::Asset::FilePile",
   "WebGUI::Asset::Wobject::Article",
   "WebGUI::Asset::Wobject::Collaboration",
   "WebGUI::Asset::Wobject::Collaboration::Newsletter",
   "WebGUI::Asset::Wobject::DataForm",
   "WebGUI::Asset::Wobject::Calendar",
   "WebGUI::Asset::Wobject::EventManagementSystem",
   "WebGUI::Asset::Wobject::HttpProxy",
   "WebGUI::Asset::Wobject::Navigation",
   "WebGUI::Asset::Wobject::Matrix",
   "WebGUI::Asset::Wobject::Poll",
   "WebGUI::Asset::Wobject::Product",
   "WebGUI::Asset::Wobject::ProjectManager",
   "WebGUI::Asset::Wobject::SQLReport",
   "WebGUI::Asset::Wobject::Search",
   "WebGUI::Asset::Wobject::Survey",
   "WebGUI::Asset::Wobject::TimeTracking",
   "WebGUI::Asset::Wobject::WeatherData",
   "WebGUI::Asset::Wobject::MultiSearch",
```

```
  "WebGUI::Asset::Wobject::StockData",
  "WebGUI::Asset::Wobject::SyndicatedContent",
  "WebGUI::Asset::Wobject::InOutBoard",
  "WebGUI::Asset::File::ZipArchive",
  "WebGUI::Asset::Wobject::WSClient",
  "WebGUI::Asset::Wobject::SQLForm"
 ],
```

After

```
"assets" : [
  "WebGUI::Asset::Snippet",
  "WebGUI::Asset::Redirect",
  "WebGUI::Asset::FilePile",
  "WebGUI::Asset::Wobject::Article",
  "WebGUI::Asset::Wobject::Collaboration",
  "WebGUI::Asset::Wobject::DataForm",
  "WebGUI::Asset::Wobject::Calendar",
  "WebGUI::Asset::Wobject::Navigation",
  "WebGUI::Asset::Wobject::Poll",
  "WebGUI::Asset::Wobject::Search",
  "WebGUI::Asset::Wobject::WeatherData",
   "WebGUI::Asset::Wobject::StockData"
 ],
```

You can see how the second list is about half the size.

In addition to the config file changes, you need to set up a preload.exclude file. To do that, run the following commands:

```
cd /data/WebGUI/sbin

cp preload.exclude.example preload.exclude
```

Now edit preload.exclude to list all of the modules that you won't be using. An example list is provided for you, and in many situations it is good enough. At the very least it will get you a good start on reducing your memory footprint.

After making these changes you must restart mod_perl for the changes to take effect.

WebGUI Cache

WebGUI has a built in caching system that helps speed up slower operations. The WebGUI cache can write out to the file system, or to a database. It depends how big your site is as to which one will perform faster. If you have a relatively small brochure-ware site, then the FileCache will outperform the database cache. However, if your site has forums, or lots of content, then the database cache will be the winner. WebGUI defaults to using the file system based cache. To switch to the database cache engine, edit your WebGUI config file and change the following directive:

```
"cacheType" : "WebGUI::Cache::Database",
```

You'll need to restart mod_perl after making this change.

If you are running multiple sites, each can have their own cache type that is better suited to that particular site's needs.

If you are using multiple load balanced web servers, you must use the database cache, as the file system cache will cause errors in a load balanced environment.

WebGUI Logging

Most people don’t believe us when we tell them this, but logging can have a significant impact on your performance.

For example, did you know that each line of information written to your WebGUI log file is 16 times more expensive than each line not written to your log file? For example, let’s say that with WARN turned on you’d log 15 lines of data over some amount of requests, and bumping it up to INFO

level will cause you to log an additional 15 lines. It's the same number of requests, either way, but you nearly double your log performance hit. Let's say it takes 1 millisecond (ms) to skip a line to be logged, then it takes 16 ms to write it out to the filesystem. That means with WARN enabled it will take 256 ms to write your log, and at the INFO level it will take 480 ms.

Granted, you're not talking about a lot of time here, but every last drop counts when you're on the web. More importantly, if you were logging at the DEBUG level, that line count might jump from 30 lines at INFO to 500 lines. That means you're at 8000 ms, for the same number of requests.

Now take that example further, and write to a remote logging server or remote database server. Not only do you have the security risk that the log stream might be intercepted by hackers and used against you, but you may have just doubled or quadrupled the number of milliseconds each log write costs.

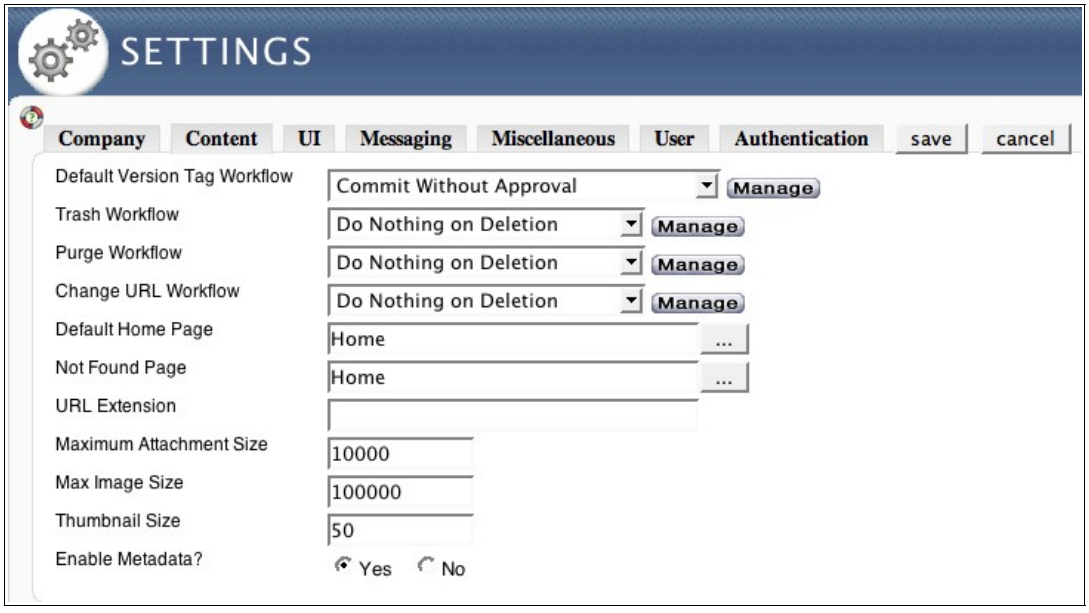
In a production environment, it's often best to log only what you absolutely need to log, and skip the rest. Log directly to your local hard disk, and then grab the logs later if you need to process or warehouse them.

Settings

The Settings screen allows you to set global site settings, such as workflows, languages, and rich editor configurations. To access the Settings screen, click on the Settings icon in the Admin Console.

Upon entering the Settings screen the default view is the Company tab. This tab lists your company's name, email address, and URL.

The Content Tab



- Default Version Tag Workfbw: this workflow determines the default workflow assigned to version tags created on the site.
- Trash Workfbw: this workflow determines what happens when

assets are placed in the Trash.

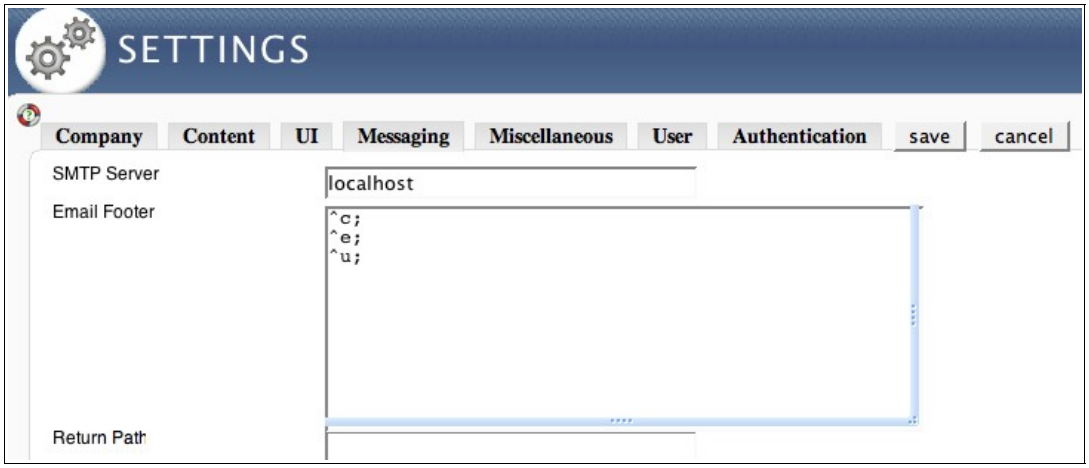
- Purge Workflow: this workflow determines what happens when assets are purged.
- Change URL Workflow: this workflow determines what happens when an asset's URL is changed.
- Default Homepage: you can set the default homepage for your site to any existing page in your site. This is the page users will be directed to when they type in your site's URL, or when they click on the homepage link generated by an AssetProxy of a Navigation Asset.
- Not Found Page: if a user requests a page that does not exist in your site, this is the page the user will be directed to. You can set this page to whatever you want, such as an error page or the homepage.
- URL Extension: allows you to add an extension, such as “php,” “html,” or “asp” to the end of your site's URL. Do NOT include the “.” (dot).
- Maximum Attachment Size: the size (in kilobytes) of the maximum attachment size allowed to be uploaded to your site. Due to the nature of HTTP Protocol, 100MB is the maximum size you can expect to upload via WebGUI's interface.
- Max Image Size: if images are uploaded to your site that are larger than the maximum allowed size, the image will be resized to conform to this limit. The max image size is measured in pixels, and the longest side of an image is used to determine if an image meets this size criteria.
- Thumbnail Size: if an image is uploaded to your site it will automatically have a thumbnail image generated at the size specified here (unless overridden on a case by case basis). Thumbnail sizes are measured in pixels.
- Enable Metadata?: if set to Yes, metadata will be enabled and allowed to be assigned to assets and tracked by WebGUI.

UI Tab

- Default Rich Editor: this is the Rich Editor configuration that will be used by default in assets that use a rich editor. This can be overridden in certain assets, such as a Collaboration System.
- Text Box Size: this determines the number of characters that can be displayed in text boxes throughout the site.
- User Function Style: determines which style to be used to style WebGUI operations (jprofiling, message log, etc) when they are available to a user.
- Admin Console Template: the style used by the Admin Console.

Messaging Tab

The Messaging tab is where you set up email service for your site.

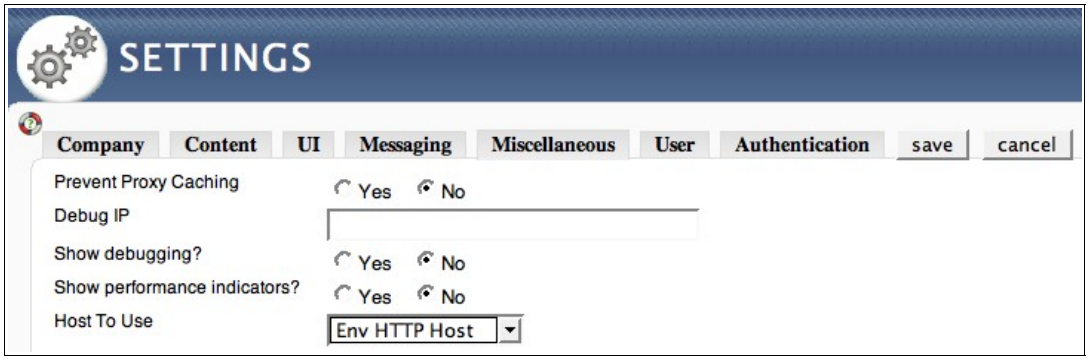


- SMTP Server: this is the address of your local mail server. It is needed for all features that user the Internet email system (ie: password recovery system). Optionally, if you are running a sendmail server on the same machine as WebGUI you can specify a path to your sendmail executable. On most Linux systems this can be found at “usr/lib/sendmail”.

- Email Footer: this footer will be processed for macros and attached to every email sent from this instance of WebGUI.

- Return Path: determines the email address to which undeliverable email messages will be sent.

Miscellaneous Tab



- Prevent Proxy Caching: some companies have proxy servers that cause problems with WebGUI. If you are experiencing problems, and you have a proxy server, you may want to set this field to Yes. Bear

in mind that this will make WebGUI's URL's less user-friendly.

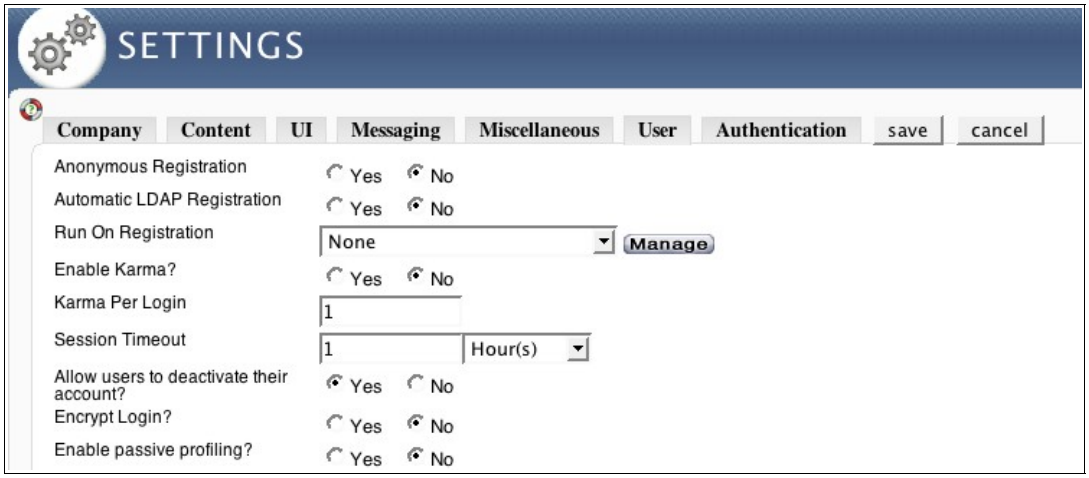
- Debug IP: this will limit debugging and/or performance output to a specific IP address or IP range. Enter the subnet in which you want to be able to view debug output in CIDR format. Multiple CIDR addresses may be entered, separated by a comma.

- Show debugging?: setting this to Yes will show debugging information in WebGUI's output. This is primarily used for WebGUI developers, but is also helpful to Administrators for troubleshooting.

- Show performance indicators?: if set to Yes, this will show the time, in seconds, it took to build each item on the page. This is helpful for debugging performance problems.

- Host To Use: determines the default host to use when generating URL's. Config Sitename will use the “sitename” variable from your config file. Env HTTP Host will use the “HTTP_HOST” variable provided by the web server.

User Tab

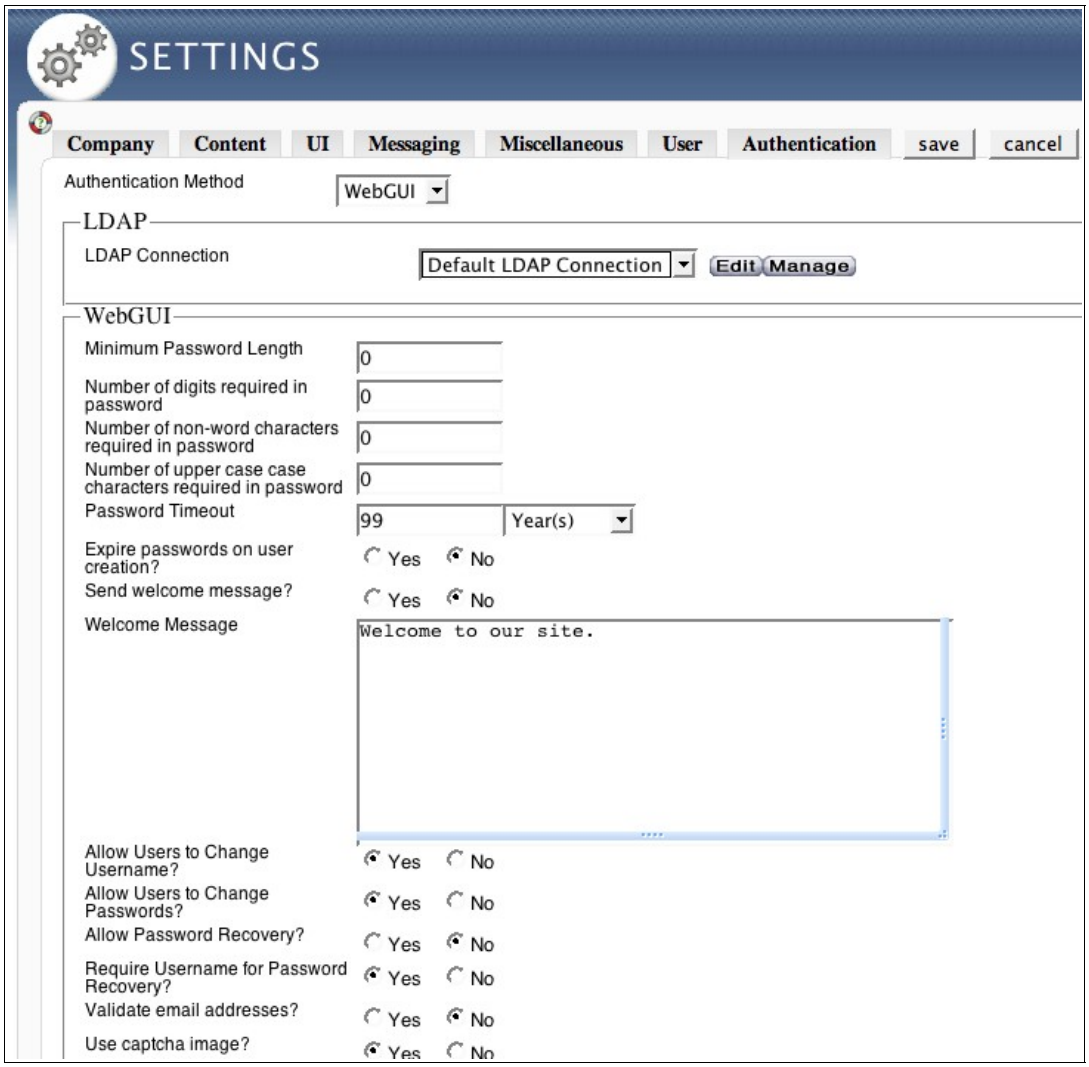


- Anonymous Registration: if toggled to Yes, users will be allowed to register themselves.

- Automatic LDAP Registration: if set to Yes, a WebGUI user account will be created and the user logged in if the credentials entered are validated by LDAP.

- Run On Registration: a workflow chosen here will be run each time a user anonymously registers.
- Enable Karma?: if set to Yes, karma will be enabled on the site. Karma is further explained in the “Karma” chapter.
- Karma per login?: if karma is enabled, this is the amount of karma a user will gain each time s/he logs in.
- Session Timeout: the amount of time a user's session remains active before that user will need to login again; this amount of time is reset each time a user views a page.
- Allow users to deactivate their account?: if set to Yes, users will be provided a means to deactivate their accounts without your intervention.
- Enable passive profiling?: if set to Yes, passive profiling will be enabled on your site. This will keep track of every asset viewed by users. Be aware, passive profiling is illegal in some areas of the world, and strongly looked down upon in others. Please be aware of the legal policy in your area.

Authentication Tab



- Authentication Method: this sets the default authentication method for new accounts in WebGUI. By default, the two available options are WebGUI and LDAP. Alternative methodscan be used by writing a customized authentication plug-in.
    - WebGUI authentication means that users will authenticate against the username and password stored in the WebGUI database.
    - LDAP authentication means that users will authenticate against an external LDAP server.

Authentication methods can be set up on a per user basis.

- LDAP Connection: Choose a connection that was defined in the LDAP Connections screen in the admin console.
- Minimum Password Length: the minimum number of characters users are required to have in their passwords.
- Number of digits required in password: the number of digits users are required to include in their passwords.
- Number of non-word characters required in password: the number of special, or non-word, characters users are required to have in their passwords.
- Number of upper case characters required in password: the number of upper-case characters users are required to include in their passwords.
- Password Timeout: the amount of time that will pass before a user is required to reset his/her password.
- Expire passwords on user creation?: if set to Yes, a password will be expired, or made unavailable for use, to any other user after this user's account creation.
- Send welcome message?: if set to Yes, a user-defined welcome message will be sent to new users who create an account.
- Welcome Message: in this field, type in the welcome message to be emailed to new users upon account creation.
- Allow Users to Change Username?: if set to Yes, users will be provided a method to change their username.
- Allow Users to Change Passwords?: if set to Yes, users will be provided a method to change their passwords.
- Allow Password Recovery?: if set to Yes, users will be given a means by which to recover their passwords should they forget them. This will be done by making portions of a user's profile available for the user to fill in, thus validating the user's identity and initiating password recovery.
- Require Username for Password Recovery?: if set to Yes, a user

must know his/her username in order to recover a password. This is advised; otherwise, someone needs only to know the other information from the user's profile to reset that user's password.

- Validate email addresses?: if set to Yes, users must validate their email address(es) before a new password will be sent.
- Use captcha image?: This requires a captcha image to be used at account creation time. Captcha is an image containing a word and some other interference to make the image hard to read by computers, but relatively easy to read by humans.
- Account Template: the template used to display users' account information.
- Create Account Template: the template displayed for account creation.
- Expired Password Template: the template used to inform a user that his/her password has expired and it is time to set a new one.
- Login Template: the template used to display the login fields.
- Password Recovery Template: the template used to display the password recovery screen.

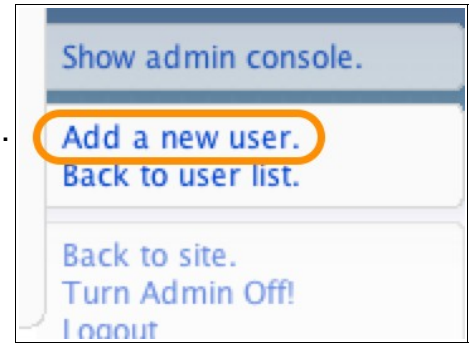
Users

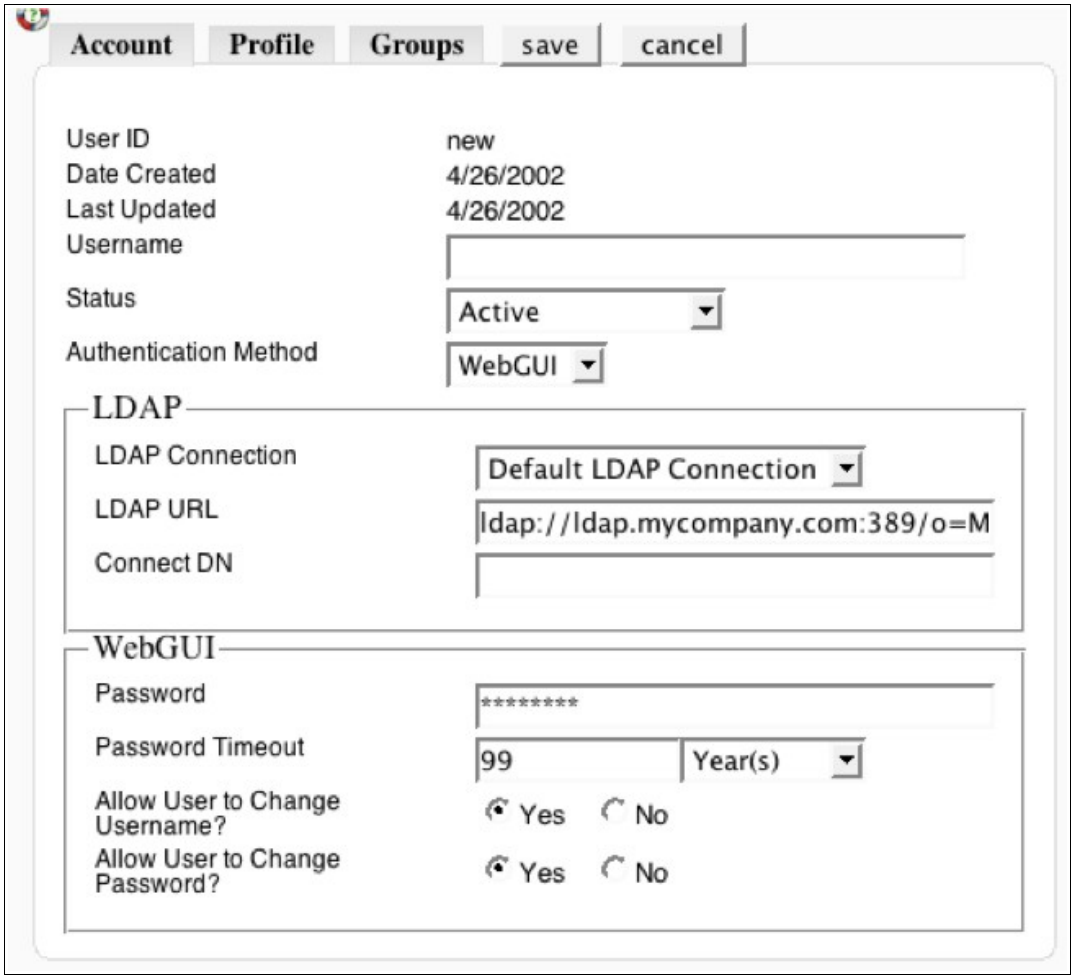
WebGUI is a user-oriented application. Although it can be used to publish static content, WebGUI is most powerful when used for user-driven applications. In order to facilitate rich user-driven applications, WebGUI has powerful user subsystems. This chapter will show you how to maximize the power WebGUI provides for user management.

Adding A User

Follow these steps to add a user.

1. Log in and go into admin mode.
2. Select “Users” from the admin bar.
3. Select “Add a new user” from the context menu on the right side of the screen.
4. You’ll now see the add user screen. From here the important fields for you to fill out are “Username” and “Password” under the “WebGUI” section.
5. Click “save” when you’re done.





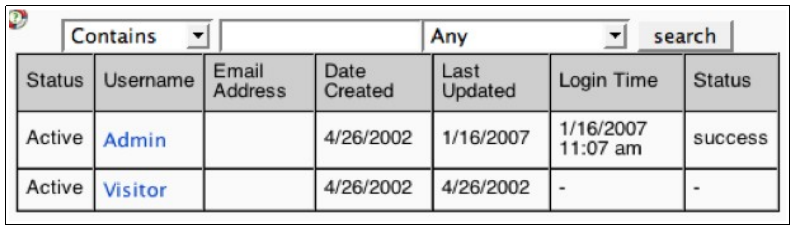
Note that there are, by default, two authentication methods, and therefore two authentication sections. They are “WebGUI” and “LDAP”. See the “LDAP” chapter for more information about configuring LDAP as an authentication source.

There are many other things you can set about a user from this screen. From the “Profile” tab you can set all the users' demographics and contact information, as well as their preferences. The Profile tab is covered in more detail later in this chapter under the headings *User Profiling* and *UI Levels*. From the “Groups” tab you can add and remove users from as many groups as you like. Groups are covered in more detail in the chapter called “Groups”.

Editing A User

To edit a user follow these steps:

1. Log in and go into admin mode.
2. Select “Users” from the admin bar.



If you have fewer than 250 users then WebGUI will display them all in a paginated list. But if you have more than 250 you'll have to search in order to view a subset of your users.

3. Now you will see the user list. From here either select a user, or search for a user to edit. You may search on any part of the username, alias, or email address.
4. Once you’ve selected a user the editing process works the same as adding a user. See *Adding A User* for details.

Configuring Authentication

WebGUI has a pluggable authentication system allowing you to expand it to allow your users to authenticate in whatever fashion they are accustomed to. Out of the box WebGUI supports two authentication methods: WebGUI and LDAP.

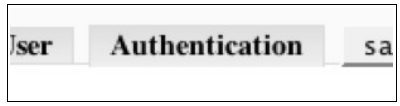
WebGUI

The WebGUI authentication method allows you to store all your users' authentication information directly in WebGUI, without having to connect to any external resources. Because it has no external dependencies, it is the easiest to use and is therefore the default option.

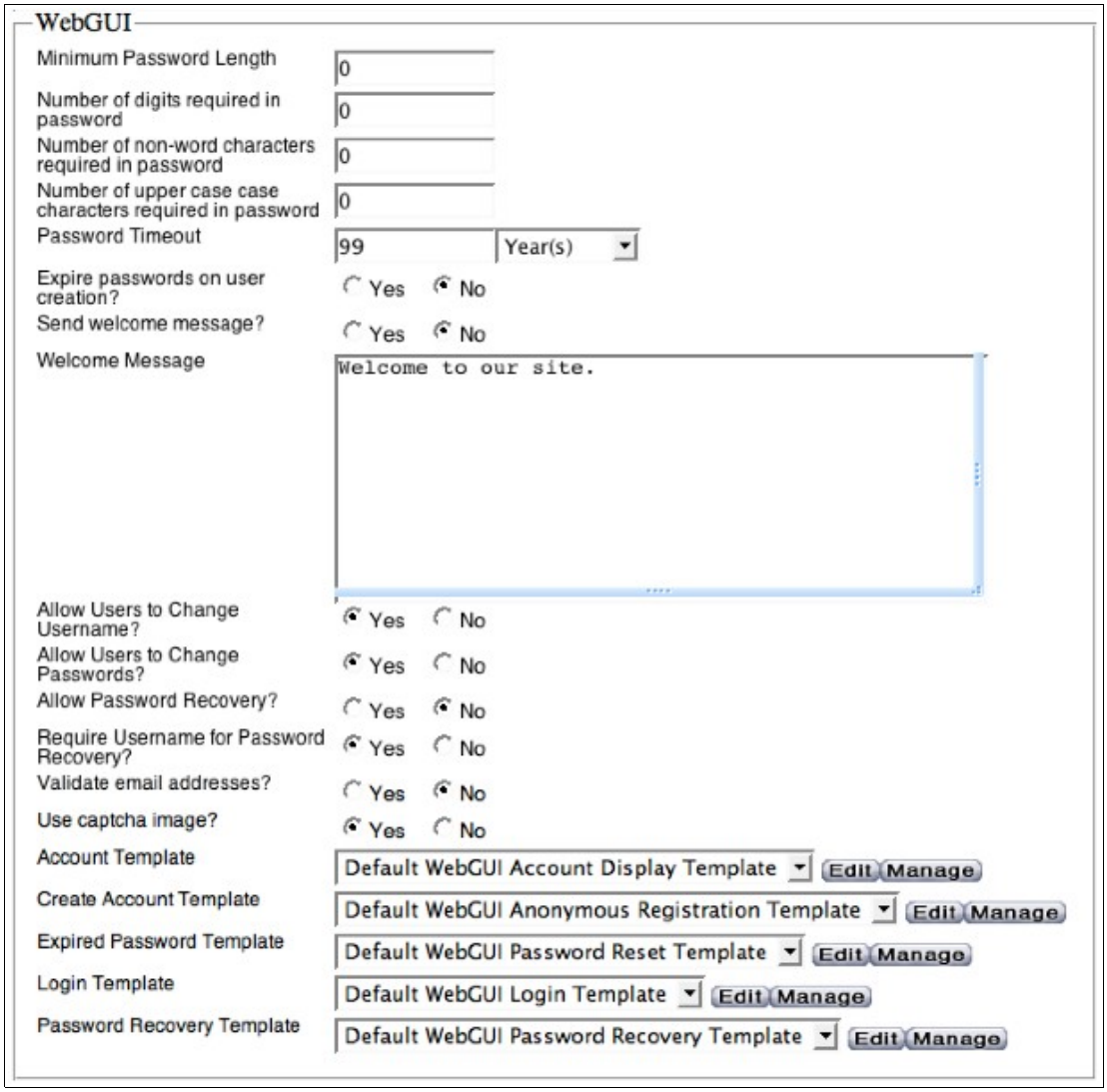
To access the WebGUI authentication configuration settings follow these steps:

1. Log in and go into admin mode.
2. Select “Settings” from the admin bar.

3. From Settings screen choose the “Authentication” tab.



4. Now you’ll see the WebGUI authentication properties. WebGUI allows for multiple simultaneous authentication methods, allowing one user to authenticate against one method, and another user to authenticate a different way.



5. There are several password options that allow you to control how your users create and change their passwords.

The options presented in these properties change as new features are added to the WebGUI authentication system.

6. You're also able to set up the templates for account related functions.

7. Click "save" when you're done.

LDAP

WebGUI also allows your users to authenticate against a directory server like Microsoft Windows Active Directory, Novell eDirectory, OpenLDAP, or Fedora Directory Server via the LDAP protocol. More information about this can be found in the chapter called “LDAP”.

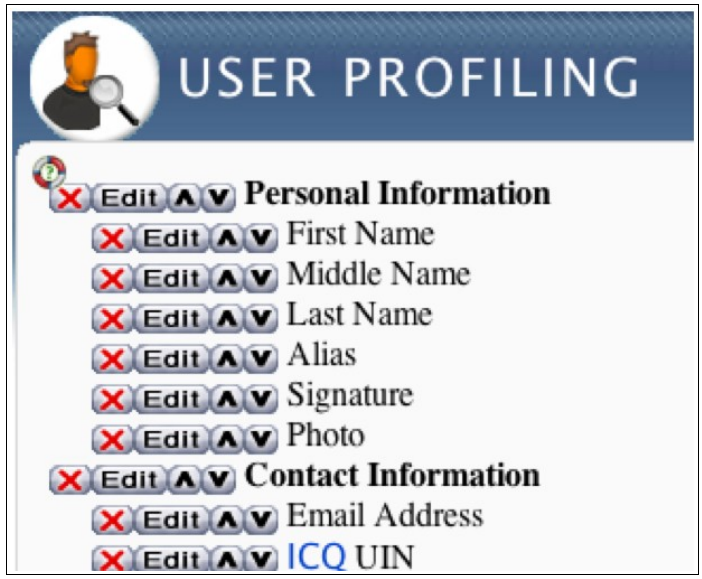
User Profiling

The user profile system allows you to create an unlimited number of fields and categories of fields for your users' accounts. In this way you can attach arbitrary data to your users. It can also be used to gather information from your users, as you can make some fields a requirement of registration.

Creating Fields

To create a field follow these steps:

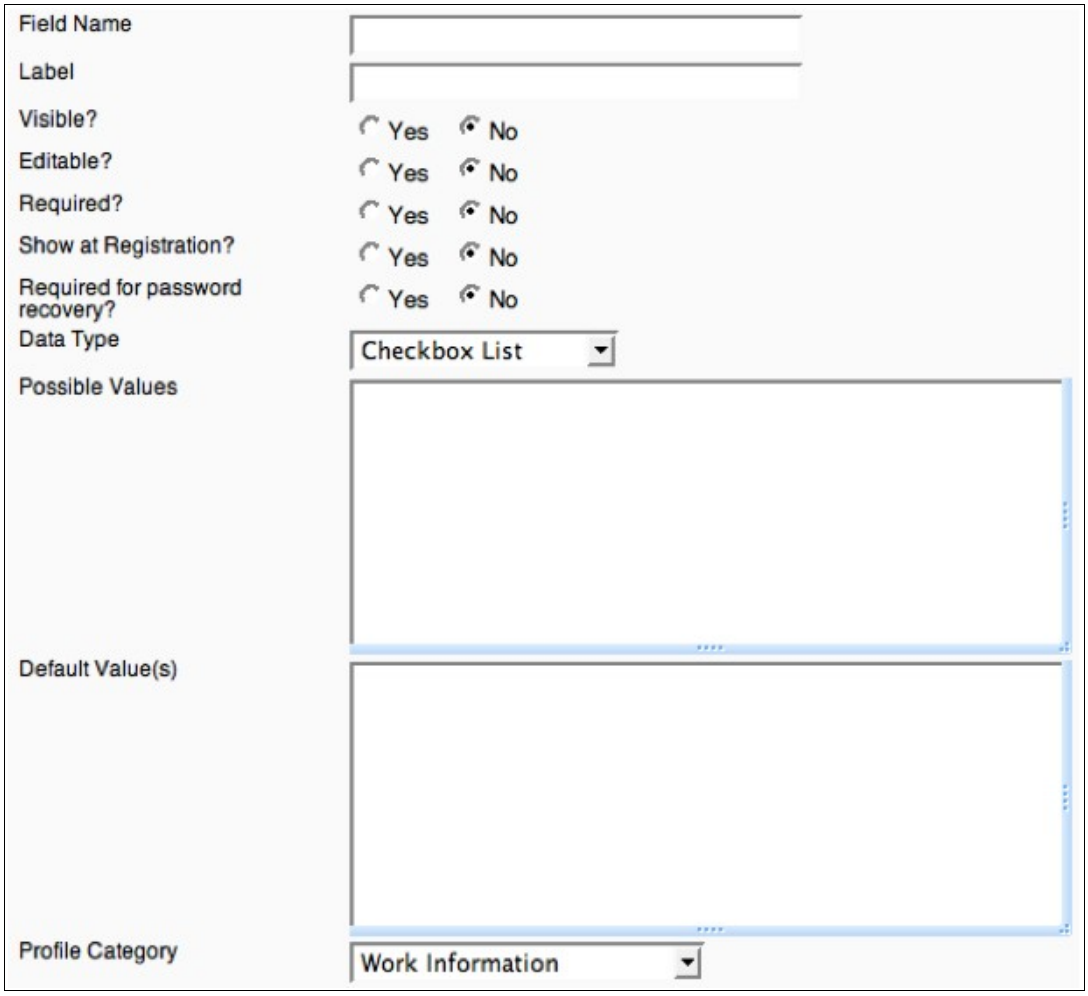
1. Log in and go into admin mode.
2. Select “User Profiling” from the admin bar.
3. Now you will be presented with a list of the fields and categories already configured in the system. Note that many common fields and categories are configured out of the box so that you don’t have to do it yourself.
4. To add a field select “Add a profile field.” on the right side of this screen.



5. You will now be presented with a list of options for creating this new field. The fields are all documented in the online help. Covered here are a specific few that people struggle with.

6. The “Field Name” field is what the field will be called behind the scenes. It should only contain letters and numbers with no spaces or special formatting.

7. The “Label” field is expecting a little bit of Perl code. You can either give it a string like: ‘My Custom Field’, or if you’re using the internationalization system to translate the label you can put in a call to it like this: WebGUI::International::get(‘my custom field’,’MyCustomModule’);



8. The “Possible Values” field is only used for list field types like “Select List”. In this field you provide a Perl hash reference of the possible values. Here’s an example of part numbers and product names. For example, if the profile field was favorite shoe:{ ‘shoe001’ => ‘Reebok’, ‘shoe002’ => ‘Nike’, ‘shoe003’ => ‘Wolverine’, ‘shoe004’ => ‘Doctor Martins’}

9. The “Default Value(s)” field is also expecting a little Perl code, but it depends upon what type of field is selected as to what the Perl code should look like. On most field types it’s expecting a string like:

   ‘Chicago’

   However, if the field type is a list, as in the example above, then it will be expecting an array reference like:

['shoe001']

And if the field is capable of multiple values, then you can even give it multiple values in the array reference like this:

['shoe001', 'shoe003']

Note that if you enter invalid code into these fields you can cause the user management functions of WebGUI to stop working. Therefore, you are only recommended to add or modify profile fields if you understand Perl. Seek professional Perl help if you have questions.

Creating Categories

Categories are used to subdivide the list of profile fields into manageable segments. To create a category follow these steps:

1. Log in and go into admin mode.
2. Select "User Profiling" from the admin bar.

3. Now you will be presented with a list of the fields and categories already configured in the system. Note that many common fields and categories are configured out of the box so that you don't have to do it yourself.
4. To add a category select "Add a profile category." on the right side of this screen.

Setting Defaults

It is common and logical to think that setting the "Default Value(s)" field when creating a field will actually set the default value of that field site wide for all users. In reality, if a user doesn't have a profile property set, then WebGUI inherits this property from the "Visitor" user. All users are based upon the Visitor user. Therefore, to set the default value for a given profile field, you need to edit the value in the Visitor user's profile. See *Editing A User* earlier in this chapter for details.

Basing the default values for the profile field off of the Visitor user instead of the Default Values in the profiling system is a side effect of how the user profiling system has historically worked. It will likely be changed in some future release.

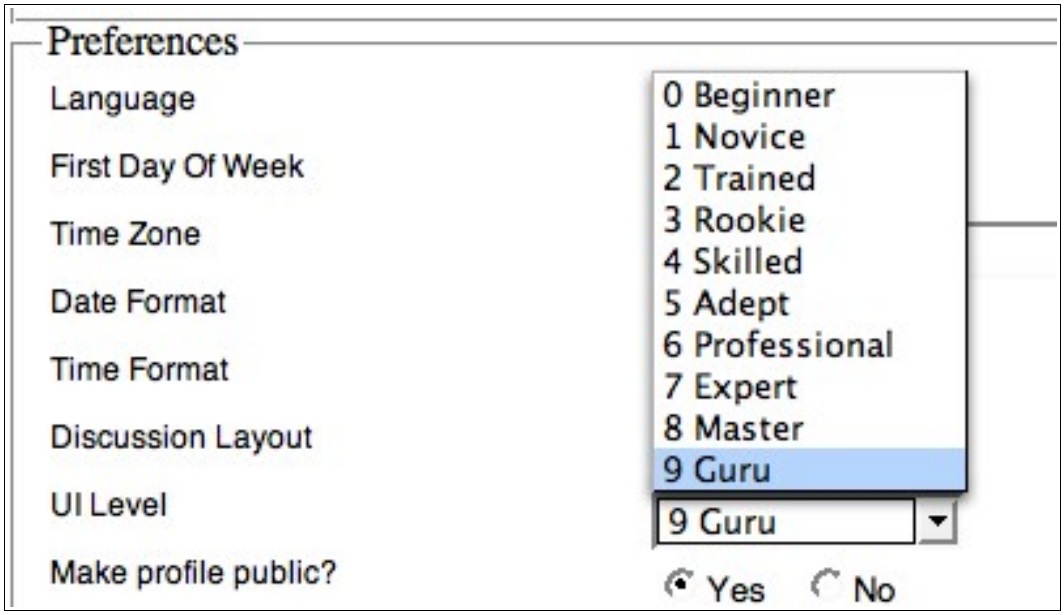
UI Levels

The User Interface (UI) level in WebGUI allows you to customize the user interface appearance and functionality on a per user basis. This is convenient if you have users who will be managing content on the site, but should not have privileges to set viewing and editing privileges or assign metadata. It's also a useful way to slowly grant users more editing power as s/he becomes more acquainted with WebGUI.

Setting A User's UI Level

A user's UI level is set in the user profile screen.

1. In the Admin Console, select Users.
2. The Users screen will open. In this screen, perform a search for the user whose UI level you'd like to set.
3. Click on the user's username in the main Users screen to enter that user's user profile. The Edit User screen will open.
4. In the Edit User screen, click on the Profile tab.
5. Near the bottom of the Profile tab is the Preferences area which contains the UI Level field.



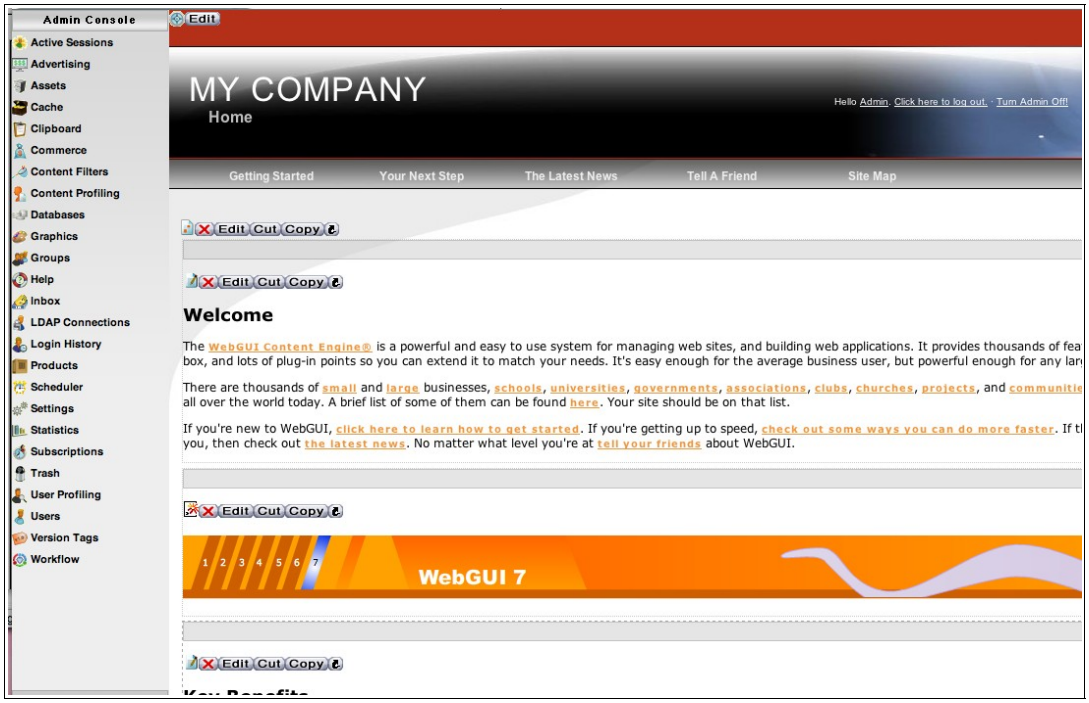
6. In the UI Level dropdown menu, select the UI Level you'd like to assign to this user, based on the user's WebGUI expertise. Level 9, Guru, will grant the user full functionality, while a lower level will offer less functionality.

7. Click save.

See examples below of sample UI Levels and how they alter the view of WebGUI in Admin Mode. The users in the sample screenshots have the ability to turn admin on and manage content.

UI Level 9: Guru

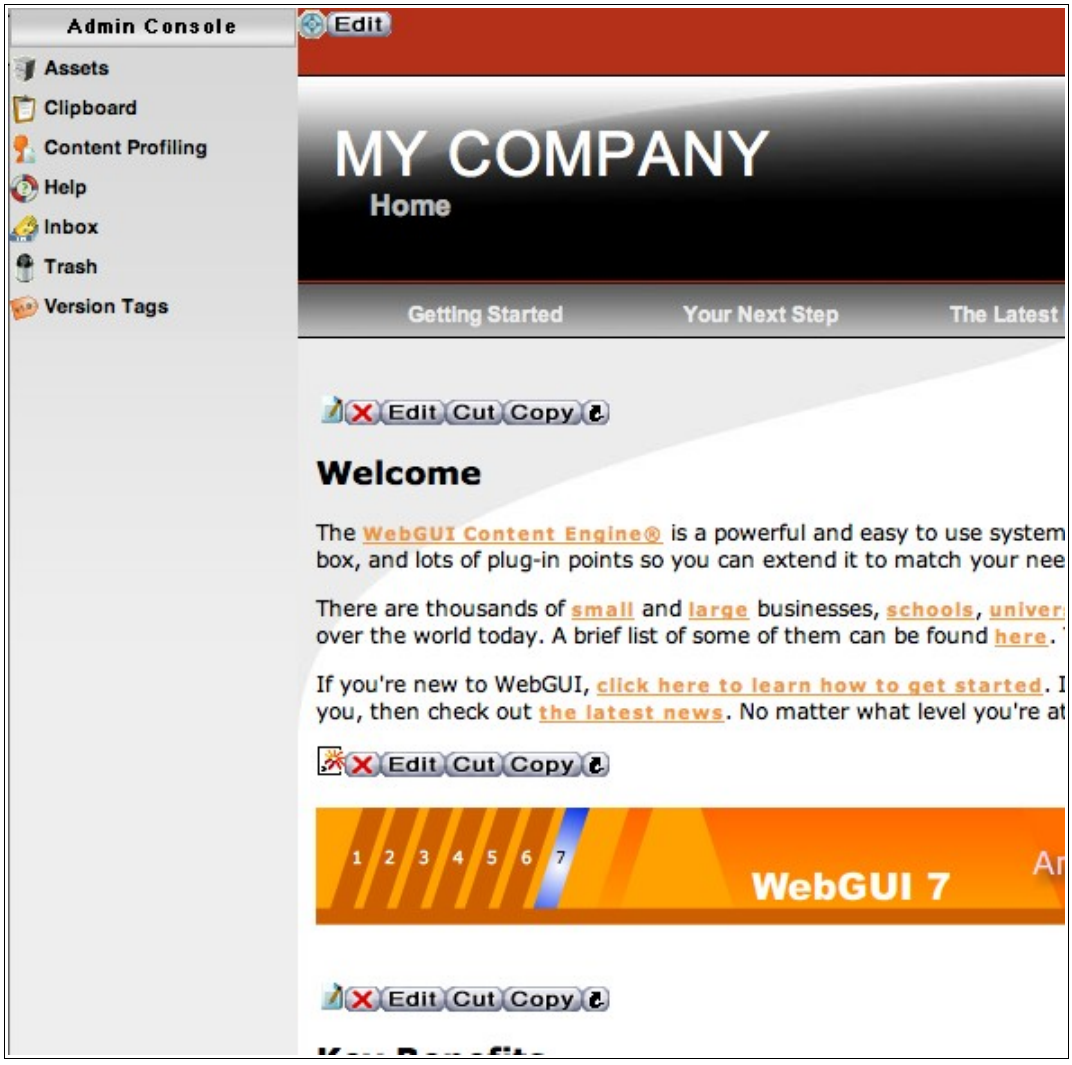
View of webpage in Admin Mode:



All options in the Admin Bar and the asset toolbars are made available. When adding an asset, all tabs are present:

UI Level 5: Adept

A view of a web page in admin mode:

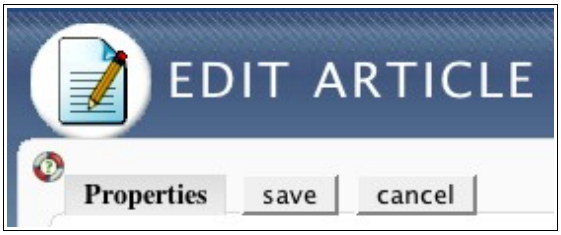


In UI Level 5 you can see that many of the options available in the Admin Console have been removed. If this user attempts to add a new asset to the site through the New Content menu, s/he will be shown an error stating that s/he doesn't have the appropriate privileges. The user is allowed to edit an existing asset; however, the Security tab has been removed from the user interface.

UI Level 1: Novice

Once again, you can see many options from the Admin Console have been removed; upon reviewing the New Content menu you would also find that some assets, such as Folder, have been removed. The shortcut arrow has also been removed from the asset toolbar.

This user can edit an existing asset by clicking on the Edit button of an asset toolbar; however, the only tab available is the Properties tab.



Overriding UI Levels

Though the assets in WebGUI have predefined UI levels set by the developer, you as an administrator can override the UI levels all the way down to the field level. You do this in the config file for each site.

Using the assetToolbarUiLevel directive you can set the UI level for each item in the asset toolbar. So, if you wanted all but the most basic functions to be hidden from view for most users you could set it like this:

```
"assetToolbarUiLevel" : {
   "edit" : 1,
   "delete" : 1,
   "cut" : 1,
   "copy" : 1,
   "shortcut" : 9,
   "editBranch" : 9,
   "lock" : 9,
   "export" : 9,
   "changeUrl" : 9,
   "promote" : 9,
   "demote" : 9,
   "manage" : 9,
```

```
   "revisions" : 9,

   "view" : 1

   },
```

Though the highest UI level is 9, you can actually set these values higher than that. If you set something to 10, it will never be seen by any user at all.

Using the assetUiLevel directive you can set the UI level for an entire asset type. Therefore, the user won't see the asset in the list of assets they can add unless they have the requisite UI level. Here's an example:

```
"assetUiLevel" : {

        "WebGUI::Asset::Wobject::WSClient" : 9,

        "WebGUI::Asset::Wobject::SQLReport" : 7,

        "WebGUI::Asset::Wobject::SQLForm" : 9,

        "WebGUI::Asset::RichEdit" : 4

},
```

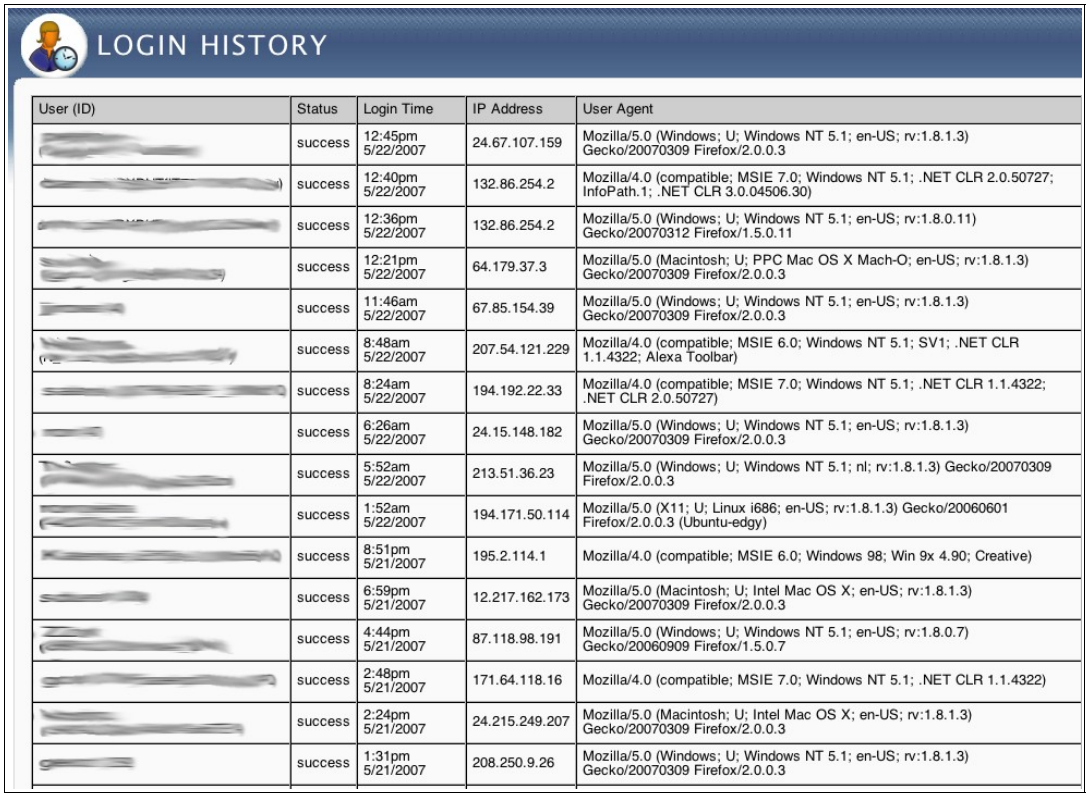
You can also set the UI level of an individual field. This is a somewhat clumsy process that will likely be replaced with something better in the future. Here's an example:

```
"WebGUI_Asset_Wobject_Article_uiLevel" : { "menuTitle" : 9, "url" : 8 },

"WebGUI_Asset_RichEdit_uiLevel" : { "askAboutRichEdit" : 7, "preformatted" : 3 },
```

What you see here is the class name of the asset followed by the keyword “uiLevel”, only instead of double colons separating the namespaces, an underscore is used. In the value part of the field is the individual field names and their new UI levels.

Login History

The Login History provides a list of users who have logged into your site, listed from most recent to oldest. To view your login history, select Login History from the Admin Console. This will open the Login History screen.



LOGIN HISTORY

User (ID)	Status	Login Time	IP Address	User Agent
[redacted]	success	12:45pm 5/22/2007	24.67.107.159	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	12:40pm 5/22/2007	132.86.254.2	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 2.0.50727; InfoPath.1; .NET CLR 3.0.04506.30)
[redacted]	success	12:36pm 5/22/2007	132.86.254.2	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.11) Gecko/20070312 Firefox/1.5.0.11
[redacted]	success	12:21pm 5/22/2007	64.179.37.3	Mozilla/5.0 (Macintosh; U; PPC Mac OS X Mach-O; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	11:46am 5/22/2007	67.85.154.39	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	8:48am 5/22/2007	207.54.121.229	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; .NET CLR 1.1.4322; Alexa Toolbar)
[redacted]	success	8:24am 5/22/2007	194.192.22.33	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322; .NET CLR 2.0.50727)
[redacted]	success	6:26am 5/22/2007	24.15.148.182	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	5:52am 5/22/2007	213.51.36.23	Mozilla/5.0 (Windows; U; Windows NT 5.1; nl; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	1:52am 5/22/2007	194.171.50.114	Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3) Gecko/20060601 Firefox/2.0.0.3 (Ubuntu-edgy)
[redacted]	success	8:51pm 5/21/2007	195.2.114.1	Mozilla/4.0 (compatible; MSIE 6.0; Windows 98; Win 9x 4.90; Creative)
[redacted]	success	6:59pm 5/21/2007	12.217.162.173	Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	4:44pm 5/21/2007	87.118.98.191	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.0.7) Gecko/20060909 Firefox/1.5.0.7
[redacted]	success	2:48pm 5/21/2007	171.64.118.16	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR 1.1.4322)
[redacted]	success	2:24pm 5/21/2007	24.215.249.207	Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3
[redacted]	success	1:31pm 5/21/2007	208.250.9.26	Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.3) Gecko/20070309 Firefox/2.0.0.3

Listed on this screen are the usernames of users who logged in, as well as an indication of if the login in attempt was successful. The time and date of the login is recorded as well as the IP Address of the user. In the far right hand column is the User Agent used to view the website.

Active Sessions

The Active Sessions screen displays currently active sessions on your site and allows you to kill sessions with a simple click.

1. To view active sessions, click on Active Sessions in the Admin Console.

2. The Active Sessions screen will open, listing all currently active sessions on the site. Included on this screen are the usernames of logged in users, a unique session signature, a date and time at which the users' sessions will expire, the date and time an active user last viewed a page, each user's IP Address, and an icon to kill a session.

ACTIVE SESSIONS

User (ID)	Session Signature	Expires	Last Page View	IP Address	Kill Session
[redacted]	5Ytcc7LCdxp-x98CJmcPYw	5/29/2007 12:52 pm	5/22/2007 12:52 pm	88.72.238.2	
[redacted]	yBPVIII8HzwPZCCPxY3bpw	5/29/2007 1:17 pm	5/22/2007 1:17 pm	207.44.136.25	
[redacted]	Kmrj39ge30ptzbUcFsn7eA	5/29/2007 12:55 pm	5/22/2007 12:55 pm	121.44.245.140	
[redacted]	_92f50wf6r68V6i8_FnwAw	5/24/2007 12:10 am	5/17/2007 12:10 am	155.143.202.184	
[redacted]	up3CWXm_0RrkSJvwzqEH3g	5/25/2007 8:20 am	5/18/2007 8:20 am	87.118.102.23	
[redacted]	vJGgKGBQfq8_0Ero9yC5MQ	5/28/2007 1:09 pm	5/21/2007 1:09 pm	87.67.15.161	
[redacted]	YL-8crJtON7qv56Ed-kmWg	5/25/2007 8:49 am	5/18/2007 8:49 am	82.221.74.87	
[redacted]	J2xndVy84NLILDqKoojgLw	5/29/2007 3:01 am	5/22/2007 3:01 am	62.216.21.20	
[redacted]	fWwPnVnHUZOzoj8oyZR6qw	5/29/2007 8:43 am	5/22/2007 8:43 am	193.2.208.9	
[redacted]	wfYqDoG2o869g3SNP23MtA	5/25/2007 2:40 am	5/18/2007 2:40 am	193.2.208.9	
[redacted]	OiESt2UONAeyrcM80VcQlA	5/29/2007 1:34 am	5/22/2007 1:34 am	69.59.17.181	
[redacted]	fIte6pTuGQVaU-vB2XuLNg	5/27/2007 2:14 pm	5/20/2007 2:14 pm	82.170.247.165	
[redacted]	ZzQzLrDJmV8jndFiYdNvVA	5/29/2007 11:04 am	5/22/2007 11:04 am	195.184.199.222	

3. To kill a user's session, simply click on the red X to the far right of the username whose session you would like to end.

Importing Users

Unless you're building a brand new site you probably will have a list of users you want to migrate from an old site. Even if you are creating a new site, you may have a list of users you'd like to bring in from some other system. Luckily, WebGUI comes with a command line utility to help you out. To access this utility, type:

```
cd /data/WebGUI/sbin
perl userImport.pl --help
```

```
Usage: perl userImport.pl --usersfile=<pathToFile> --configfile=<webguiConfig>

    --usersFile     File (and path) containing import information.

    --configFile    WebGUI config file (with no path info).


Options:

    --authMethod    The authentication method to be used for
                    each user. Defaults to 'WebGUI'. Can be
                    overridden in the import file.

    --canChangePass If this flag is set users will be able to change
```

```
            their passwords.  Otherwise not.

--delimiter     The string that separates each field in the
            import file. Defaults to tab.

--expireOffset  The the amount of time before the user will
            be expired from the groups they are added
            to. Defaults to the expire offset set in
            the group definition within WebGUI. May be
            overridden in the import file.

--expireUnits   Valid values are "seconds", "minutes",
            "hours", "days", "weeks", "months", "years",
            "epoch", or "fixed". Defaults to "seconds". This is
            the units of the expire offset. If set to
            "epoch" the system will assume that the
            expire offset is an epoch date rather than
            an interval.  If set to "fixed" the
            system will assume that the expireDate is
            a fixed date.

--groups        A comma separated list of group ids that
            each user in the import file will be set
            to. Can be overridden in the import file.

--help          Display this help message.

--identifier    Alias for --password.

--ldapUrl       The URL used to connect to the LDAP server
            for authentication. Can be overridden in
            the import file.

--override      This utility is designed to be run as
            a privileged user on Linux style systems.
            If you wish to run this utility without
            being the super user, then use this flag,
            but note that it may not work as
            intended.

--password      The default password to use when none is
            specified with the user. Defaults to
            '123qwe'. Can be overridden in the import
            file.
```

```
    --quiet         Disable output unless there's an error.

    --status        The user's account status. Defaults to
                    'Active'. Other valid value is 'Deactivated'.

    --update        looks up all the users from the file in the database
                      and updates all the given fields for each user that
                      exists in the database. users that are in the file
                      and not in the database are ignored.

    --updateAdd     looks up the users from the file in the database
                    and updates all the given fields for each user that
                    exists in the database. users who do not exist in the
                    database are added as new users.

    --replaceGroups when updating, if the user already belongs to some group
                        this flag will delete all the user's existing groups and
                        and the new groups to him/her


User File Format:

    -Tab delimited fields (unless overridden with --delimiter).

    -First row contains field names.

    -Valid field names:

        username password authMethod status
        ldapUrl connectDN groups expireOffset

    -In addition to the field names above, you may use any
    valid profile field name.

    -The special field name 'groups' should contain a comma
    separated list of group ids.


Special Cases:

    -If no username is specified it will default to
    'firstName.lastName'.

    -If firstName and lastName or username are not specified,
    the user will be skipped.
```

-Invalid field names will be ignored.

-Blank lines will be ignored.

-If userId is specified for an import record, that userId
be used instead of generating one.

Using this utility you can import users, assign profile fields, assign users to groups, and update the settings of existing users. To use it, you first need a file that contains users you wish to import. Here's an example:

```
username    identifier      groups      email
bob    blue42      3      bob@bob.com
jane    blue43      a3j3sk3sk4k3jDJjs_s3Ex,i1xn3sj43Kjesl5sy7Bx3    jane@bob.com
joe                 joe@bob.com
```

The first row contains the field which is going to be imported. Each field name is separated by a tab.

The second row describes a user named “bob” who has a password of “blue42”. Bob must be the boss because he's being added to a group with an id of “3”, which is the “Admins” group. Finally, there is Bob's email address. Note that these parameters don't necessarily line up with the column headings. That's ok, because they aren't columns, they're tab separated fields.

The third row describes a user named “jane” who has a password of “blue43”. Jane is being added to two groups. You can see a comma separating the two group ID's. Finally, Jane's email address is displayed.

The fourth row describes a user named “joe”. There is no password defined for Joe, which means you can either specify one on the command line, or his password will default to “123qwe”. There are also no groups specified for Joe, so he won't have any specific initial privileges. Finally, Joe's email address is shown.

To import this group of users, simply type the following command:

```
perl userImport.pl --conf=www.example.com.conf --users=/path/to/users.txt
```

You'll see some output as the import process proceeds.

```
Starting up...OK
Adding user bob
Adding user jane
Adding user joe
```

```
Cleaning up...OK
```

If you run the command a second time you'll get some output noting that you've already imported these users.

```
Starting up...OK
User bob already exists. Skipping.
User jane already exists. Skipping.
User joe already exists. Skipping.
Cleaning up...OK
```

You could also use the --update or --updateAdd command line arguments if you wanted to update something about your users, rather than skipping over them.

Locked Out?

Since the early days of WebGUI we've gotten support questions from our users who have locked themselves out of WebGUI. They've forgotten the passwords to their admin accounts.

You should create an extra account with admin access. This leaves a back door for yourself should you ever accidentally do something to lose access on the first account. Prepare for disaster before disaster strikes.

The good news is that you can get in, even if you haven't done this, by following these simple steps.

Forgot Admin Username

1. If you've changed your admin account username and forgotten what you've changed it to, then log in to the MySQL command line like this:

```
mysql -uUSERNAME -pPASSWORD www_example_com
```

2. You can get your username and password from your WebGUI config file.

3. Then type this command.

```
select username from users where userId='3';
```

4. That will return output of your username.

```
BigAdminDude
```

5. If you wish you could just change it back to “Admin” like this:

```
update users set username='Admin' where userId='3';
```

Forgot Admin Password

You cannot recover a lost password because WebGUI uses a one way encryption technique to store passwords. This ensures that if others were able to view your stored password in the database, it would be useless to them. However, if you've forgotten your admin password, you can easily reset it to the default “123qwe”. Log in to MySQL as described above, and then type this command:

```
update authentication set fieldData='RvlMjeFPs2aAhQdo/xt/Kg' where userId=3 and
authMethod='WebGUI' and fieldName='identifier';
```

Be sure to set it to something more secure after you log in.

Page/Style Template Broken

Another way you might get locked out of WebGUI is if you break a page or style template. For instance, in the style template you might forget to place the template variable that puts content onto the page. The good news is that you can log in and do other admin functions using only some simple URL's.

1. To log in, use this URL:

```
http://example.com/?op=auth;method=login;username=admin;identifier=123qwe
```

2. To turn admin on use this URL:

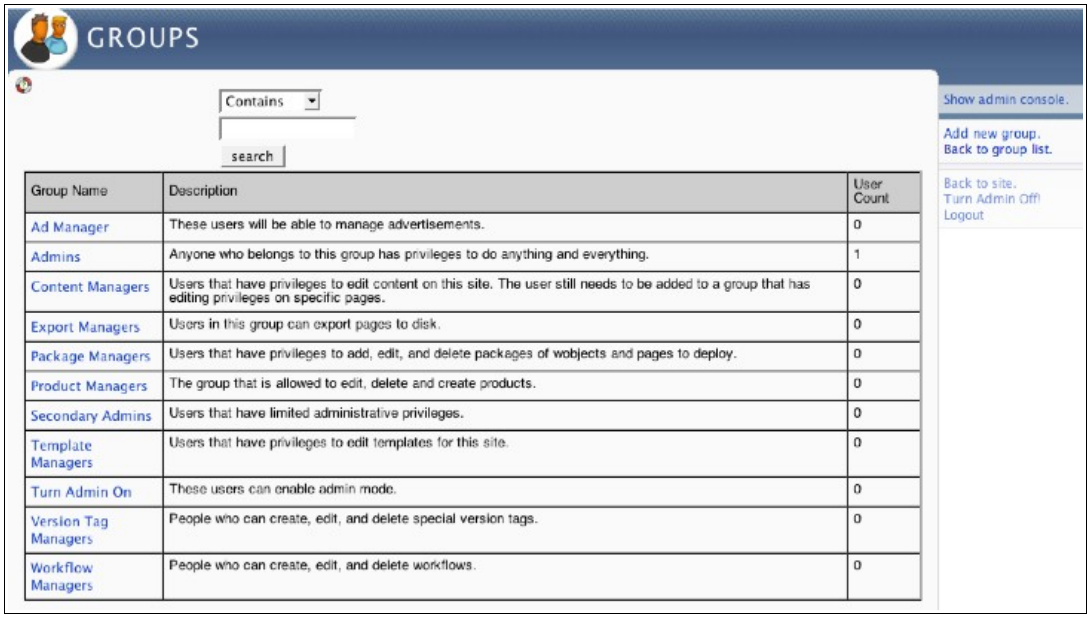
```
http://example.com/?op=switchOnAdmin
```

3. And to view the admin console go to:

```
http://example.com/?op=adminConsole
```

Groups

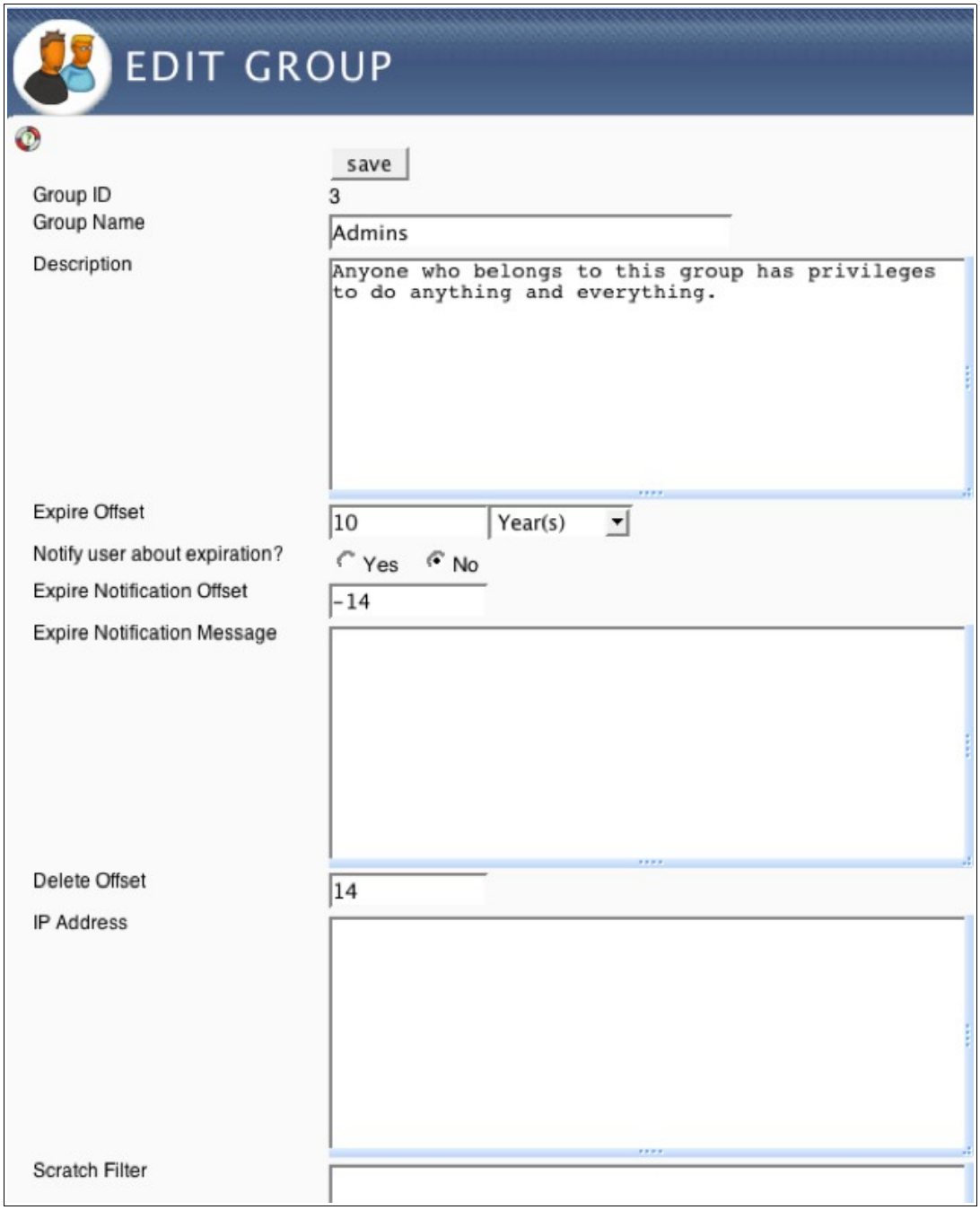
Group management in WebGUI allows you to control access to very specific sections of your website. Users are assigned to a group, and then that group is assigned viewing or editing rights to a page or asset. This is often used on subscription sites where paying subscribers are able to access content on a site that is otherwise not visible to the average visitor. You are probably familiar with this concept from the "Security" tab of add/edit asset screens. The "Who can View" field determines the group of users who may view a given asset.



Group Name	Description	User Count
Ad Manager	These users will be able to manage advertisements.	0
Admins	Anyone who belongs to this group has privileges to do anything and everything.	1
Content Managers	Users that have privileges to edit content on this site. The user still needs to be added to a group that has editing privileges on specific pages.	0
Export Managers	Users in this group can export pages to disk.	0
Package Managers	Users that have privileges to add, edit, and delete packages of wobjects and pages to deploy.	0
Product Managers	The group that is allowed to edit, delete and create products.	0
Secondary Admins	Users that have limited administrative privileges.	0
Template Managers	Users that have privileges to edit templates for this site.	0
Turn Admin On	These users can enable admin mode.	0
Version Tag Managers	People who can create, edit, and delete special version tags.	0
Workflow Managers	People who can create, edit, and delete workflows.	0

The group management feature is accessed through the Admin Console. Clicking on "Groups" will open the group management screen.

Listed on this screen are all the groups currently existing, a brief description of that group, and the number of users currently assigned to each group. Clicking on a group name will open the "Edit Group" screen.



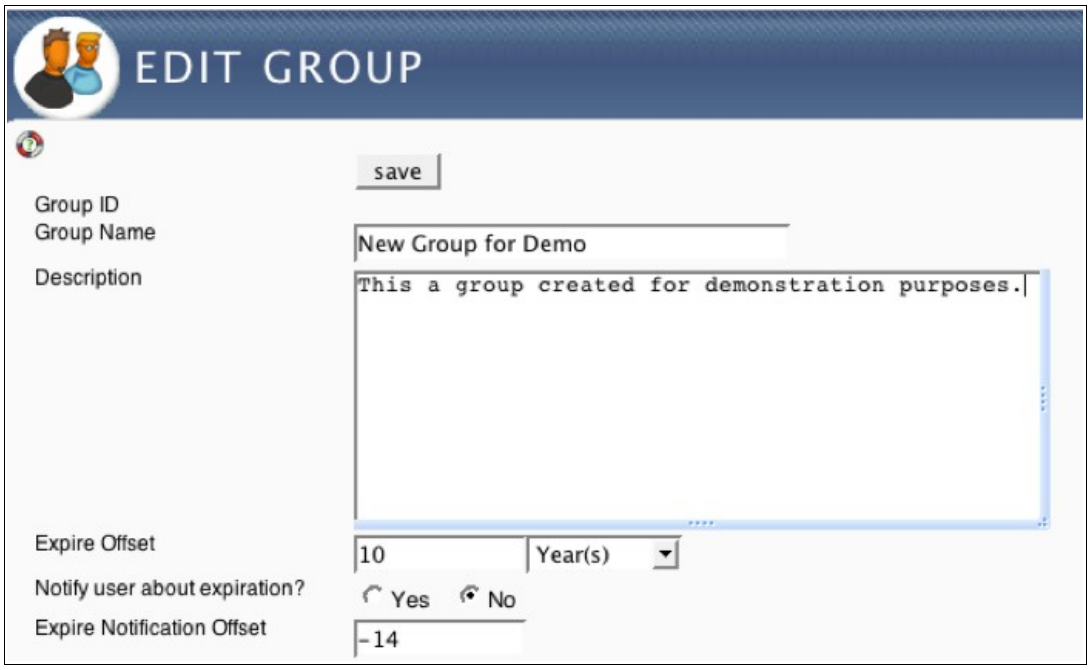
- Group Name: a name or title for this group.
- Description: a description of the what this group is for.
- Expire Offset: the amount of time that can pass before the group access expires.

- Expire Notification Offset: determines how much advance notice a group member will receive before group membership expires.
- Expire Notification Message: enter a message the user will receive via email upon group expiration.
- Delete Offset: the number of days that will pass between expiration notification and the group's actual deletion from the system.
- IP Address: if this is specified, any users visiting the site from that IP range will automatically be a member of this group. This is great for intranets where you want to grant access to some information, but only if the users are inside your network.
- Scratch Filter: users can be dynamically bound to a group by a scratch variable in their sessions.
- Users can add/remove themselves?: set if users can add or remove themselves from a group. For example, if a user clicks on a Subscribe link in a forum, the user is electing to add him/herself to that group.
- Database Link: if you'd like to have this group validate users using an external database, select the database to use.
- SQL Query: to validate users against an external database you may construct an SQL statement that will return a list of WebGUI userids for users in the group.
- The LDAP fields can be used to bind users in a group to an existing LDAP directory.
- Cache groups for how long?: large sites using external group data will make many calls to the external database. To help reduce the load, you may select how long you'd like to cache the results of the external database query within the WebGUI database.

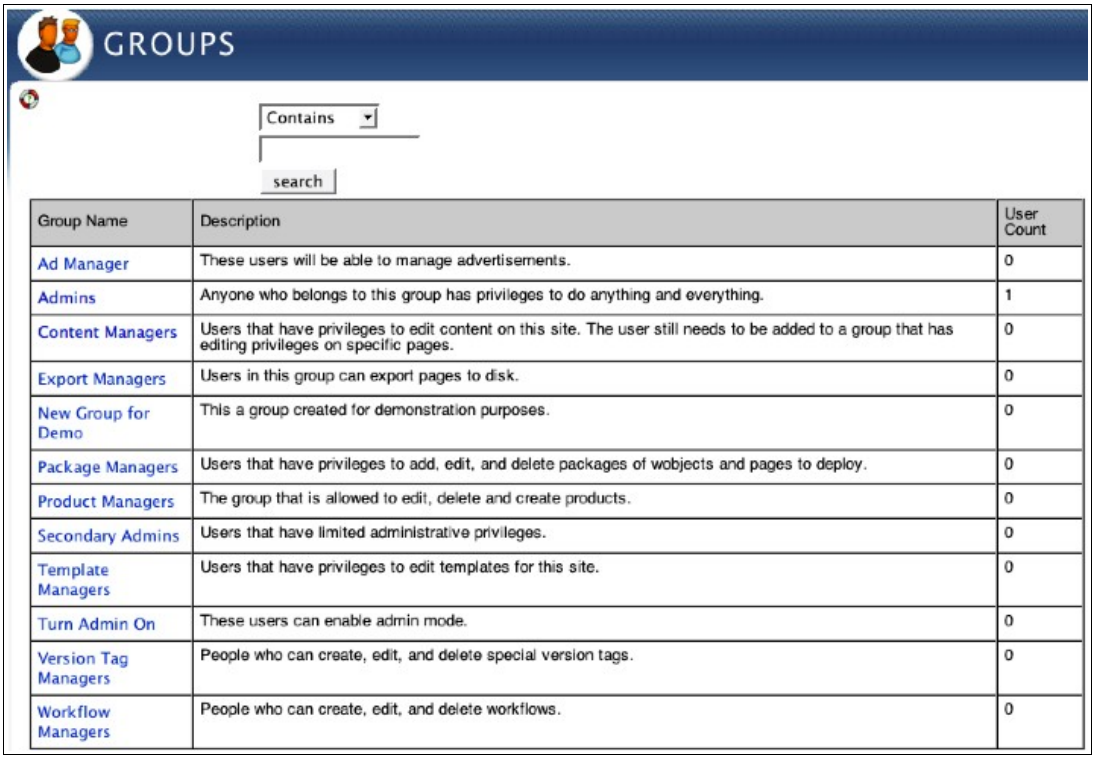
Create a Group and Add a User

To create a new group, locate the "Add new group" link on the far right hand side of the screen.

A blank “Edit Group” screen will open in which you may enter a group name and a brief description of the group’s purpose.



Click “save” at the top of the screen, and the group will be added to the list on the “Groups” screen (seen here as the fifth group listed).



GROUPS

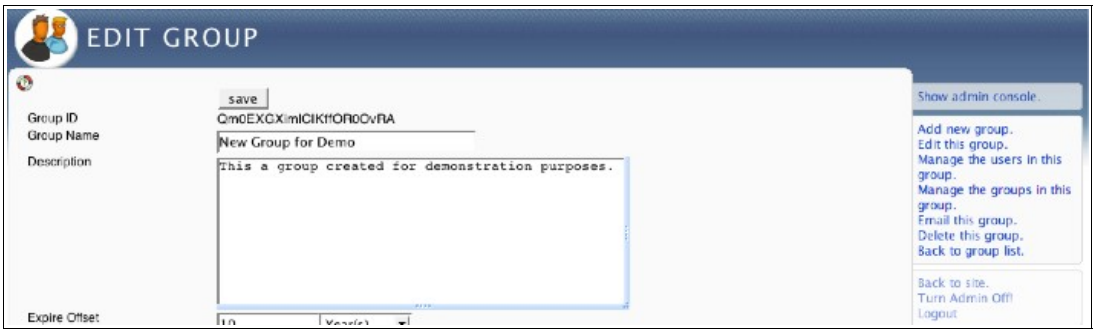
Contains

search

Group Name	Description	User Count
Ad Manager	These users will be able to manage advertisements.	0
Admins	Anyone who belongs to this group has privileges to do anything and everything.	1
Content Managers	Users that have privileges to edit content on this site. The user still needs to be added to a group that has editing privileges on specific pages.	0
Export Managers	Users in this group can export pages to disk.	0
New Group for Demo	This a group created for demonstration purposes.	0
Package Managers	Users that have privileges to add, edit, and delete packages of wobjects and pages to deploy.	0
Product Managers	The group that is allowed to edit, delete and create products.	0
Secondary Admins	Users that have limited administrative privileges.	0
Template Managers	Users that have privileges to edit templates for this site.	0
Turn Admin On	These users can enable admin mode.	0
Version Tag Managers	People who can create, edit, and delete special version tags.	0
Workflow Managers	People who can create, edit, and delete workflows.	0

Managing Users In A Group

Now that the group is created, users will need to be added to it. You can only add users that currently exist in the system to a group. To assign users to a group, begin by clicking on the group's name in the "Groups" screen. This will open the "Edit Group" screen for the selected group.



EDIT GROUP

save

Group ID Qm0EXCXimICIKffORt0OvRA

Group Name New Group for Demo

Description This a group created for demonstration purposes.

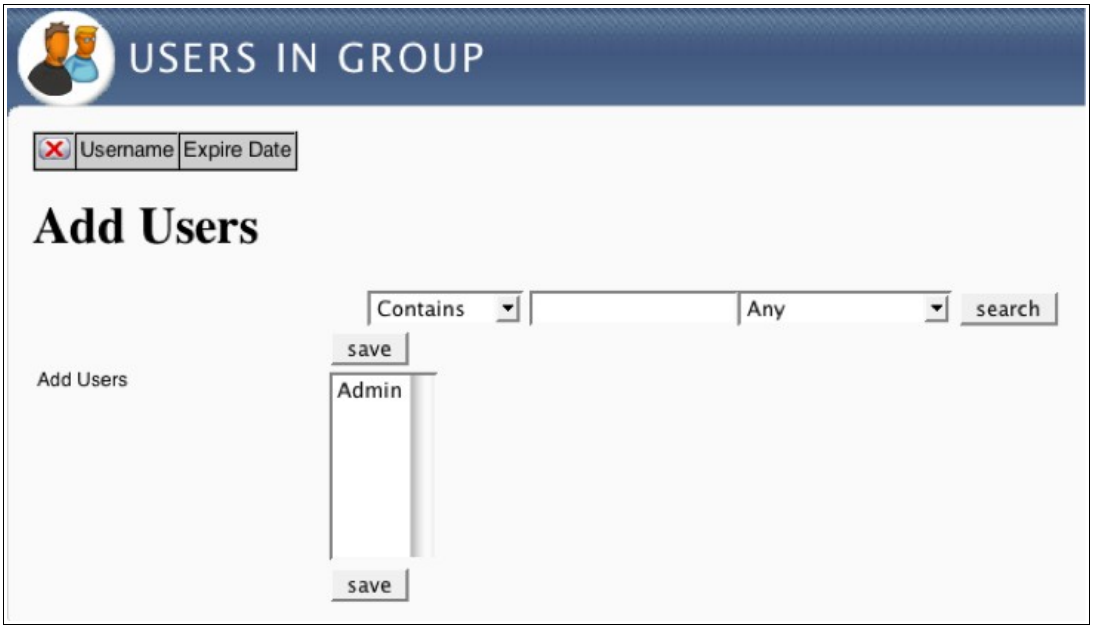
Expire Offset

Show admin console.

Add new group.
Edit this group.
Manage the users in this group.
Manage the groups in this group.
Email this group.
Delete this group.
Back to group list.

Back to site.
Turn Admin Off!
Logout

On the far right hand side of the screen, click on the "Manage the users in this group" link. This will open the "Users in Group: Add Users" screen.

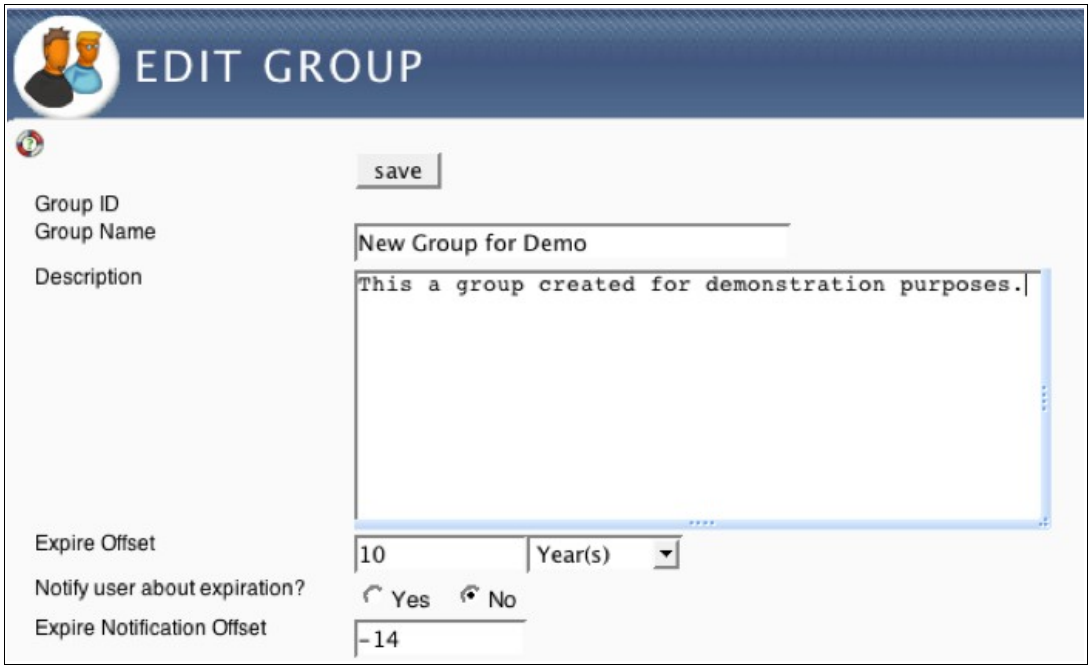


From this screen you have a couple options. If users already existed in this group, you would see them listed at the top of the screen. You can search for a user to add if you know its username. You will also see an “Add Users” field with a vertical menu of usernames to the right (currently, the only user is Admin). You can select users from this list and then click save to add them to the user list. If Admin is selected, and save is clicked, Admin is made a user in this group.

To remove a user from the group you would simply click the redX next to the username, or use the checkboxes to select multiple users to delete at one time. Users’ profiles may also be accessed by clicking on the edit button next to their usernames.

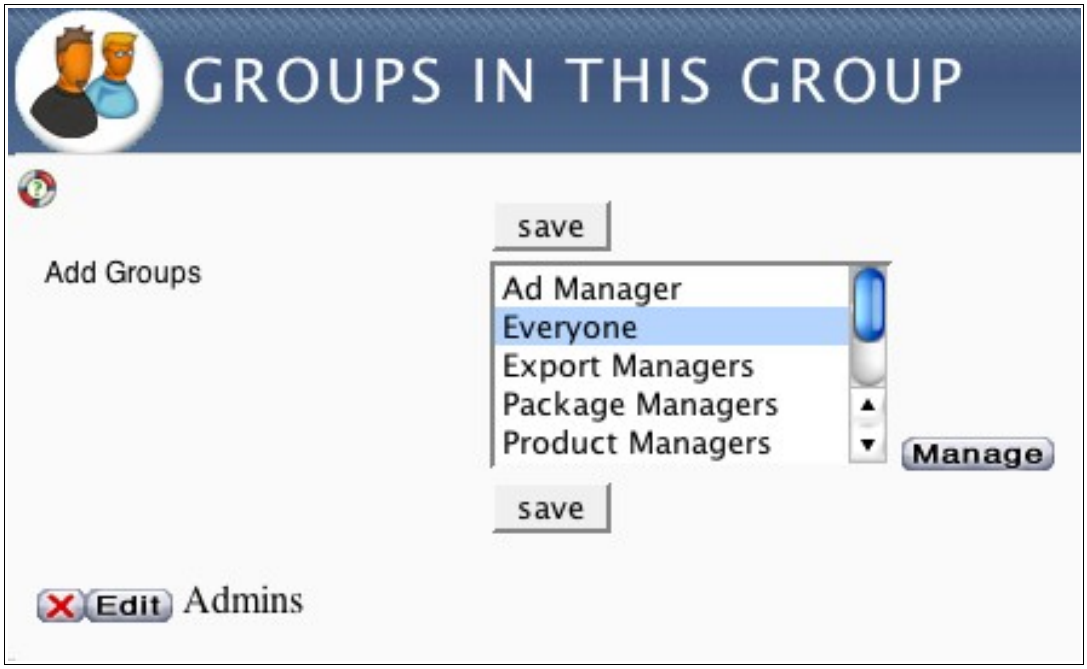
Editing A Group

To edit an already existing group, select a group from the Groups screen, and click on the *Edit this group* link located on the far right hand side of the screen. The Edit Group screen will open, containing the previous information input at the time of creation. Simply update the group information you'd like to edit, and click save.



Managing Groups in a Group

Groups can be added to other groups to efficiently include entire groups of users in one step. To do so, select a group from the Groups screen. On the far right hand side of the screen, click on the *Manage the Groups in this Group* link. This will open the Groups in this Group screen.



At the bottom of the Groups in this Group screen is a list of all groups already a member of this group. Next to each group is a toolbar. Click on the red X to delete a group from this group, or the Edit button to edit the individual group. To add a group to this group, highlight a group(s) from the “Add Groups” menu and click save. The selected group(s) will be added to the list at the bottom of the screen.

Managing Expirations

One of the more powerful features of the WebGUI groups system is that it allows you to automatically expire users from groups. This is useful for automatically removing privileges from:

- students at the end of a semester.
- limited term employees.
- clients who have purchased a term-based support agreement or subscription-based content access.

There are a number of settings that you can use to deal with group expiration.

Expire Offset is the main trigger for group expirations. It determines how long a user should be a member of a group when added to it. The offset is the amount of time that should pass from the time the user is added to the group until s/he is removed from the group. Don't worry, you can set the final expiration date on a per user basis, which will be shown in a bit.

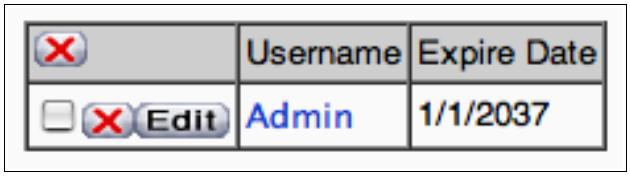
You may optionally notify the user that s/he is going to be, or has been, removed from the group by selecting yes on the “Notify user about expiration?” setting.

If you chose yes, then you can set the number of days prior to or after the actual expiration date that you'd like to notify the user. For example if you want to notify them two weeks before the expiration, you'd type “-14”. If you wanted to notify them on the day of expiration, then you'd type “0”. And if you wanted to notify them 3 days after they had already been expired then you'd type “3”.

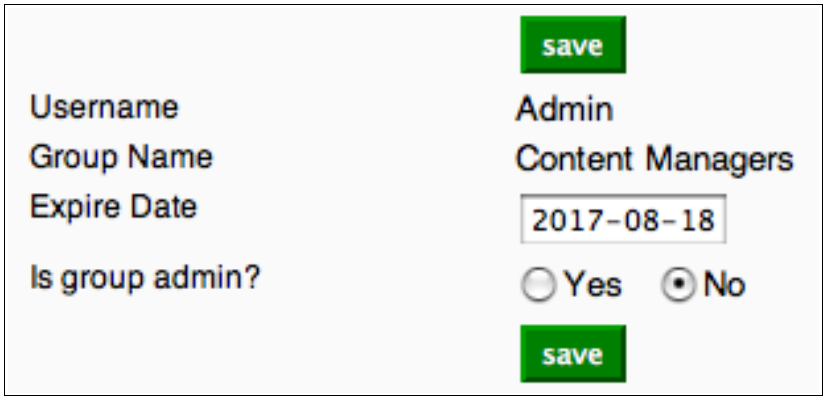
Then you can write a brief message about why they're being removed from the group. This message will then be emailed to users based on the notification offset.

The delete offset is when the users will actually be removed from the group. Note that as soon as they have been expired they no longer have any group privileges, but they are still a member of the group. This can be useful if you want to build a report of recently expired users. If expiration removed them from the group you wouldn't know that they were recently expired! In addition, if you plan to notify the users after they've already been expired, then the delete offset should be higher than the notification offset.

As previously mentioned, you can also edit the actual expiration date of an individual user in a group. To do this, you'll need to edit the grouping relationship. Go to Groups > Manage users in group and then you'll see a list of users in the group:



Now, click on the “Edit” button to edit the grouping relationship. That will bring you to a screen that looks like the following.

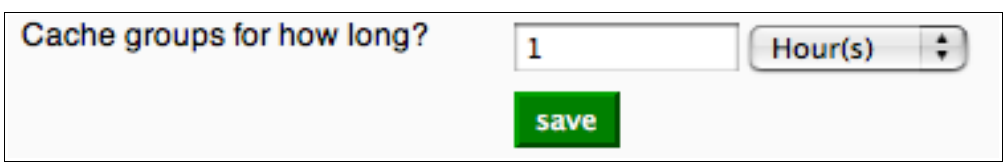


Here you can either type in the expire date, or use the calendar helper to chose it.

Special Inclusion

Another powerful feature of WebGUI's group system is special inclusion. Special inclusion means that the user is not directly part of the group, but they can be dynamically added to the group by some special rules.

How long the user remains a part of the group is up to the individual rule, as well as the cache setting. Because in some cases evaluating these rules can be performance intensive, the result is cached.



If you set the cache to last for 1 hour (which is the default), then if the special inclusion rules determine that the user is or is not part of the group, that determination will be the same for 1 hour, regardless of whether the rules have changed. If this amount of caching is not acceptable for the rules you're trying to apply, then set the cache to 1 second. On the other hand, if the result of the rule check won't change very often, then set the cache timeout to a higher value, like 1 day.

Scratch Variables

A scratch variable is a variable attached to a user's session for the duration of the session. If the user logs out, or if his/her session times out, then this variable will no longer be set.

You can use scratch variables for special inclusion. The most typical example where this would be used is if you don't want a user to have access to some information until they have accepted a license agreement or a privacy policy. Once they've accepted the agreement, then you'll give them access to whatever they were looking for.

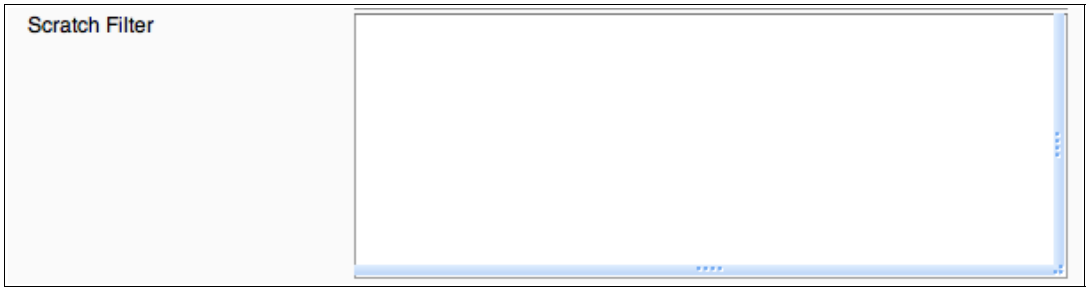
Most of the time, scratch variables are set by a programmer, so unless you are the programmer who set the variable, you likely won't know if it exists. However, you can also set scratch variables via a URL. Here's what that looks like:

```
/page?op=setScratch;scratchName=EULA;scratchValue=1
```

In the above example a variable called “EULA” has been created and assigned a value of “1”. However, to prevent trampling on scratch variables set by programmers, WebGUI automatically prepends “www_” to scratch variables set via the URL. That means the variable has been renamed to “www_EULA”.

Now that you have a variable set, you can use this in a group for special inclusion. You can set what is called a scratch filter in the group. A scratch filter looks like:

```
www_EULA=1
```



If users click on your URL, they'll automatically be added to the group. However, scratch filters can require more than just a single variable. Perhaps you want them to agree to both an end user license agreement and your privacy policy. In that case, you can create a second URL for them to click on like this:

/page?op=setScratch;scratchName=**privacy**;scratchValue=**1**

And then you can make your scratch filter look like this:

www_EULA=1;www_privacy=1

The value doesn't need to always be “1” either. For example, you may want the user to select from a list of options, and then you can display different information based upon the selection. Let's say you have a page that has information that's useful to people from Canada or the United States, but the information is unique to each of them and you don't want to display both. You can then create two groups and two URL's, and assign one group to each page or asset that contains the information related to that group.

On the landing page you'll provide URL's that look like this:

/page?op=setScratch;scratchName=whereFrom;scratchValue=Canada

/page?op=setScratch;scratchName=whereFrom;scratchValue=US

Then, you'll create your “Canada” and “U.S.” groups. In the Canada group you'll add a scratch filter like this:

www_whereFrom=Canada

In the U.S. Group you'll add a scratch filter that looks like this:

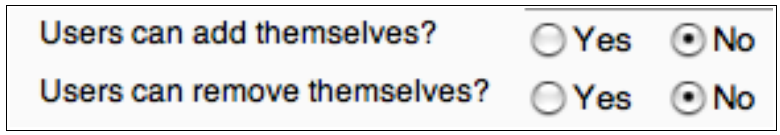
www_whereFrom=US

Now you've used the same variable to differentiate two distinct groups of users.

Autonomy

You can also allow users to autonomously add and remove themselves from some groups. This is useful for voluntary subscriptions and permanent acceptance of license agreements.

To allow or disallow this you would select “Yes” on these two options in the group settings:



You have the option of setting these parameters individually because there are several circumstances where you might want to allow a user to opt in or out, but not vice-versa. For example, once they agree to a license you may not want them to change their minds later. Or, you may want to create a group of users to send emails to, and don't want other people to add themselves to that list, but legally in most countries you do need to give them a way to opt out of such emails.

Once you have determined that users can opt in or out of a group, you have to create the links that the users can use to do that. There are two macros you can use to do just that. They are ^GroupAdd(); and ^GroupDelete();. The syntax for both is the same:

```
^GroupAdd(EULA, Click here to agree to the terms of service.);

^GroupDelete(Customer Email List, Click here to unsubscribe.);
```

Those will create links that look like:

Click here to agree to the terms of service.
Click here to unsubscribe.

You can also optionally specify a template ID as the third parameter like this:

```
^GroupDelete(Customer Email List, Unsubscribe!, Pbtmpl0000000000000041);
```

You can then create a template to lay out your opt in or opt out. Your only template variables are “group.url” and “group.text”, so it won't make for a very exciting template, other than your own content, but you can do it. In general, no template is necessary.

Database

You can tie group membership to a database query as well. Though it's not trivial, this means you could theoretically even tie privileges to applications external to WebGUI. In order to use this option, your database table must either already contain the WebGUI User ID's of the users you wish to include in the group, or you must have an external table that marries the WebGUI User IDs to the data in your other tables.

For the sake of example, let's say you have a fictional accounting system called Megabux. Let's also say that you have set up privileges for accounting department users in Megabux that you want WebGUI to respect, so that when a Bean Counter is promoted to Chief Executive Bean Counter, the new privileges are also transferred to WebGUI. And when a Stub Lackey is fired, his privileges are automatically removed from WebGUI. This is not only a plausible scenario, but one that we hear all the time. The departments, systems, and roles may change, but the problem is the same.

Megabux has some database tables in a database called DB9 that look like this:

users	
uid	integer(11)
username	varchar(20)
password	char(10
given_name	varchar(30)
sur_name	varchar(30)
user_defined	text

permissions	
uid	integer(11)
rid	integer(11)

roles	
rid	integer(11)
role	varchar(20)

The first thing you need to do is create a group in WebGUI for each role that you want to bind to in Megabux. Then, you're ready to ponder your

inclusion options.

Ultimately, what the WebGUI group is looking for is a list of WebGUI user ID's to be returned to it. So, you have to find a way to match WebGUI users to Megabux users. Depending on your needs, there are many ways to do this:

- Require that the Megabux “uid” or “username” field be created as a profile field in WebGUI.
- Put the WebGUI “userId” or “username” into the “user_defined” field in Megabux.
- Create an additional table that maps the WebGUI “userId” field to the Megabux “uid” field.
- Make sure that the WebGUI “username” field matches the Megabux “username” field.
- Make sure that the Megabux “given_name” and “sur_name” fields match the “firstName” and “lastName” fields in WebGUI's user profile.

Once you have identified how you want to match WebGUI users to Megabux users, then you have one more hurdle to overcome: synchronization. If WebGUI and Megabux both have databases in the same MySQL server then you can just do a cross database query like this:

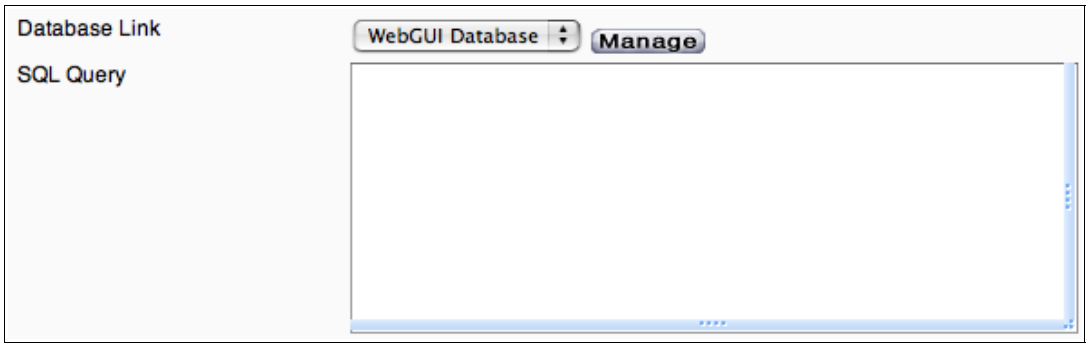
```
select
        webgui.users.userId
from
        webgui.users
left join
        megabux.users
        on megabux.users.user_defined=webgui.users.userId
left join
        megabux.permissions
        on megabux.users.uid=megabux.permissions.uid
```

```
where

        megabux.permissions.rid='133'
```

That last part, '133', represents the ID of a role in Megabux that you want to match to a WebGUI group.

However, if WebGUI is in MySQL and Megabux is in DB9, then you're going to have to write some external script that will synchronize the data using the method chosen above. Writing this script is not difficult to do, but it goes beyond the scope of this book. It would be enough to say that the script would simply connect to both databases once per hour or per day (depending on how often things change) and synchronize the data used to match a WebGUI userId to a Megabux uid.

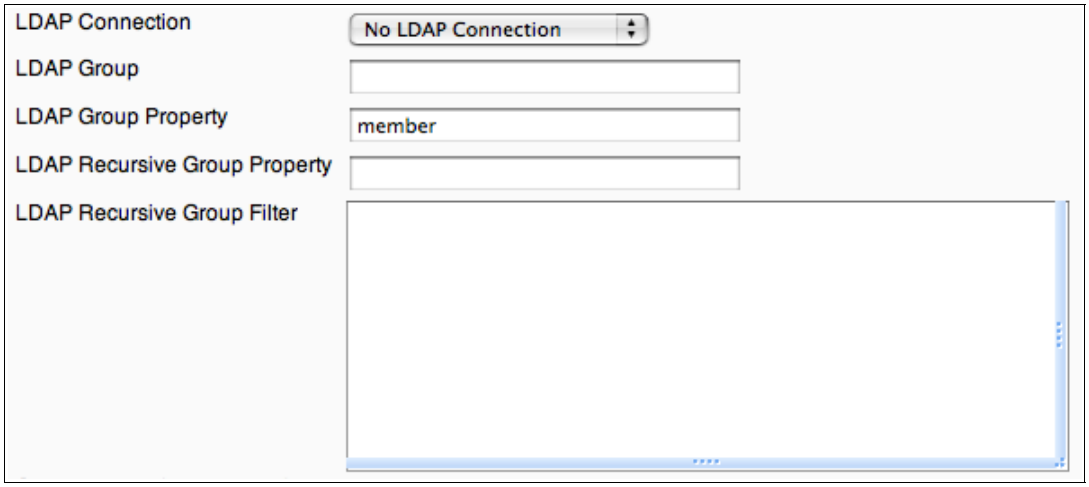
Once you have all of the above mechanisms sorted out, the actual group settings are easy. The form looks like this:



Choose a database link to connect to. See the chapter on Database Links in this book if you're unfamiliar with the topic. Then, type the SQL query into the field. Remember that it needs to return multiple rows of WebGUI userIds and nothing else.

LDAP

Similar to a relational database, you can use LDAP for special inclusion into WebGUI groups.



Select an LDAP connection from the list of defined connections. Note that in order for this to work, the users that you're trying to get a list of must also log in via this LDAP connection. This can get very complicated very quickly if you have users connecting from more than one LDAP server. If you are unfamiliar with LDAP connections see the chapter on LDAP for a better understanding.

In the LDAP Group field, type the full distinguished name (DN) of the group, like this:

cn=My Group, ou=groups, dc=example, dc=com

In the Group Property field, provide the attribute that the group members are listed under. On most LDAP systems this is “member”.

In the Recursive Group Property field, provide the attribute that other groups are listed under that should also be searched recursively for users. Sometimes this is also “member”, as in Group A is a member of Group B. However, this attribute is implemented differently on each system, and not at all on some systems. Consult you LDAP server's documentation for details.

The Recursive Group Filter is a way to speed up the recursive search system by automatically eliminating some results without recursing through them. It's just a simple string matching system, but if your “member” attribute includes both users and groups, but a group will never contain a property called “uid”, then it can speed up queries immensely.

IP Address

WebGUI also gives you the ability to use a user's IP address for special inclusion. There are two main instances where this is interesting:

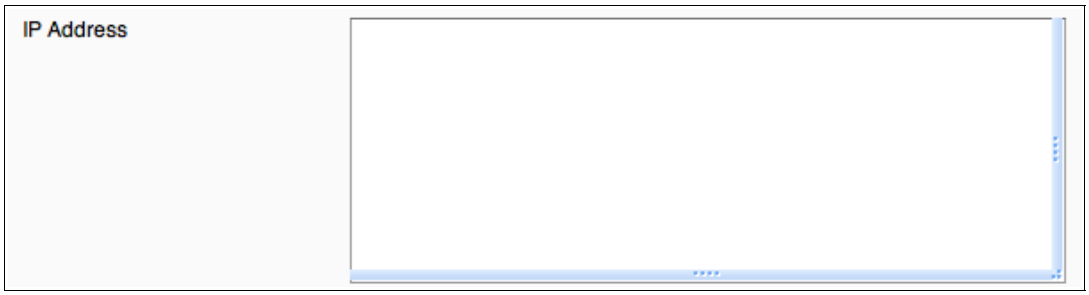
- If you want to give your employees access to certain things while they are in the office, but not allow anyone to see those things outside the office.

  Be careful with this setting. It is possible for a hacker to spoof an IP address.

  - A sub case of this is that you only want administrators to be able to access the user management system or groups system while inside the office. See the chapter on “Settings” for details on giving privileges to admin functions.

- You know your user's IP address or IP range, but you do not know which user accounts belong to him/her. In this way you can give a user access to resources on your site while s/he is coming from the known IP address or IP range.

Enter an IP address or range using CIDR notation. You may also add multiple addresses if they are separated by commas.



Here's an example:

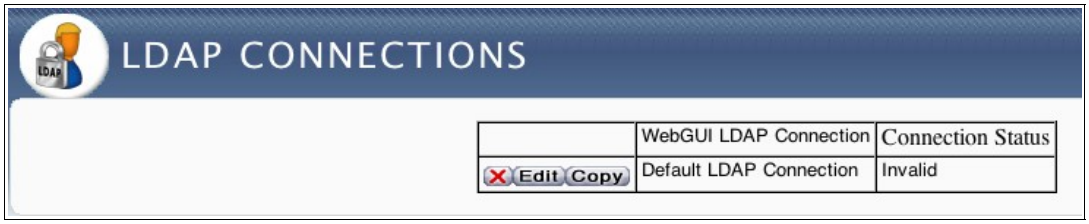
10.0.0.1/32, 192.168.1.0/24

That will add anyone coming from 10.0.0.1, or anyone whose IP address starts with 192.168.1., into the group during the cache period.

LDAP

WebGUI allows you to connect to an LDAP source for both authorization and authentication. These LDAP sources can also be used in your own custom programs for things like workflow routing if your LDAP server contains information on employee hierarchy.

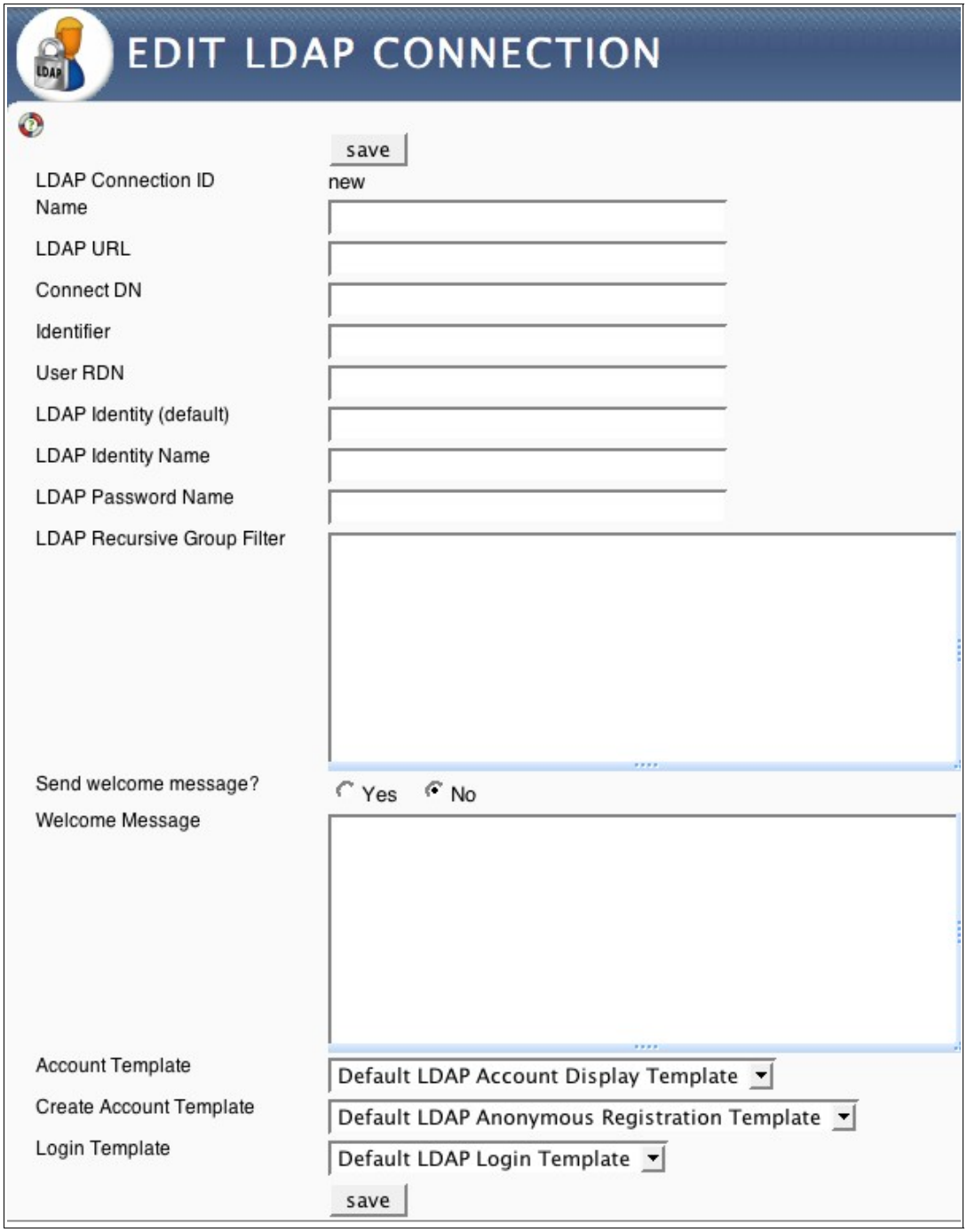
The main LDAP screen under the Admin Console looks like this:



Here, you can see all of your existing LDAP sources. One particularly interesting thing is the connection status column. This tells you whether this source has been configured properly as to whether WebGUI can connect to it.

To add a new source click on the “Add an ldap connection.” link.

You will then be presented with a screen containing all the properties you can assign to an LDAP connection.



Background Information

Though many people think of LDAP as a storage mechanism, it is actually a network protocol. So when you hear someone talking about connecting to

an “LDAP Server”, that means that the protocol you're using to connect to the server is LDAP, but the storage mechanism on the server is called a “Directory”.

Some server documentation refers to directory protocols as the X.500 specification. LDAP is actually a subset of X.500. You don't need to concern yourself with X.500 because almost no one actually uses it. LDAP was created because X.500 was too heavy and slow. Just remember that if you see X.500 it's most likely talking about LDAP.

Configuring an LDAP Source

Configuring an LDAP Source in WebGUI is fairly simple.

1. First, in the admin bar choose “LDAP Connections” under “Admin Console”.

2. Then, click on the “Add an LDAP connection.” link on the right side of the screen.

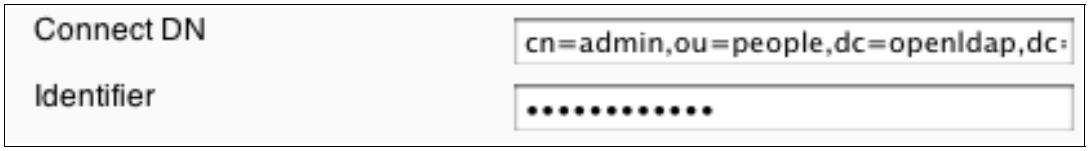
3. Enter a name for your server and its URL.

4. If your LDAP server allows anonymous read access (most don't) then you can click save now. Otherwise, you also need to specify some account information that WebGUI can use to connect to the server for administrative functions.

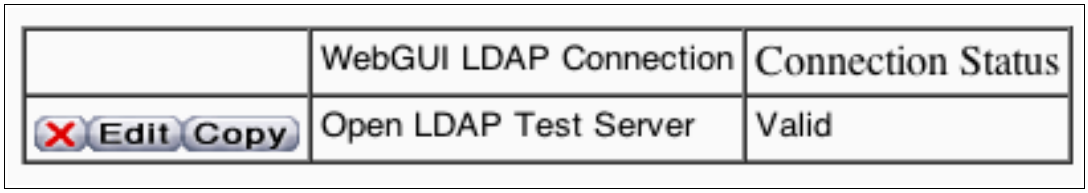
   A. In the Connect DN field put in the fully distinguished name or DN

for an account that has full read access to the directory. Most admins choose to make an account specifically for this purpose.

B. In the Identifier field, type in the password associated with the user from the Connect DN.



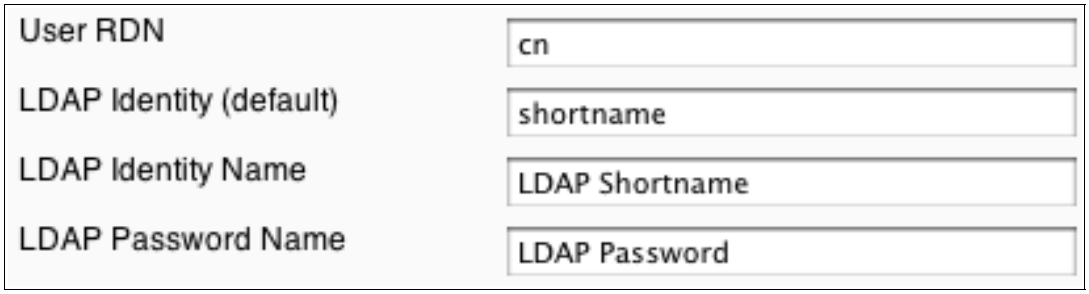
5. Now hit save so that WebGUI can check to see if it has a valid connection to your LDAP source. You should see something similar to the following if it can connect to your LDAP source.



Authentication

Now that you have an LDAP connection you probably want to do something with it. The most common use for LDAP is authentication. By using LDAP for authentication your users can use the same usernames and passwords everywhere throughout your organization.

To set up authentication edit your LDAP connection and fill in the following fields:



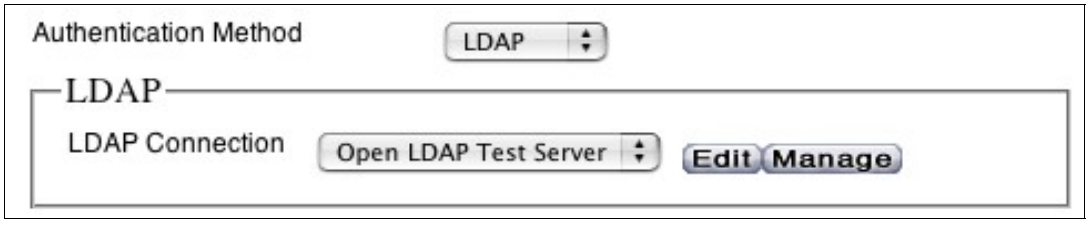
- User RDN, or Relative Distinguished Name, is just the first part of the user's distinguished name. This is almost always “cn”.

- LDAP Identity (default) is the element in your directory that WebGUI will use to look up a user to find out its distinguished name. This element must be unique and it varies from system to system. Commonly, the field will be “shortname”, “username”, or “uid”. This is typically what the user will use for a username when authenticating against your LDAP source.

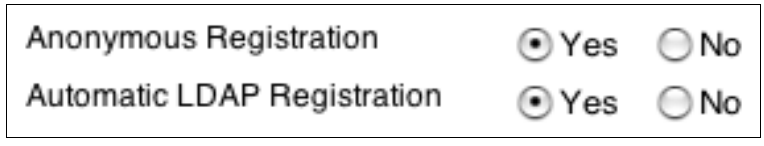
- LDAP Identity Name and LDAP Password Name are just human readable labels, so you can put whatever you want in those fields. Sometimes admins will just use “Username” and “Password”, but often their users already know these fields by other names, so they use something like “Email Username” or “Windows Username” or “Internet Username”.

Now that you've configured the LDAP source for user authentication, you need to tell WebGUI to use it. To do this go to Admin Console > Settings and then to the Authentication tab.

Set the Authentication Method to “LDAP” and choose the LDAP connection we just set up.



You're not quite done yet. Now go to the User tab and choose your settings for Anonymous Registration and Automatic LDAP Registration:



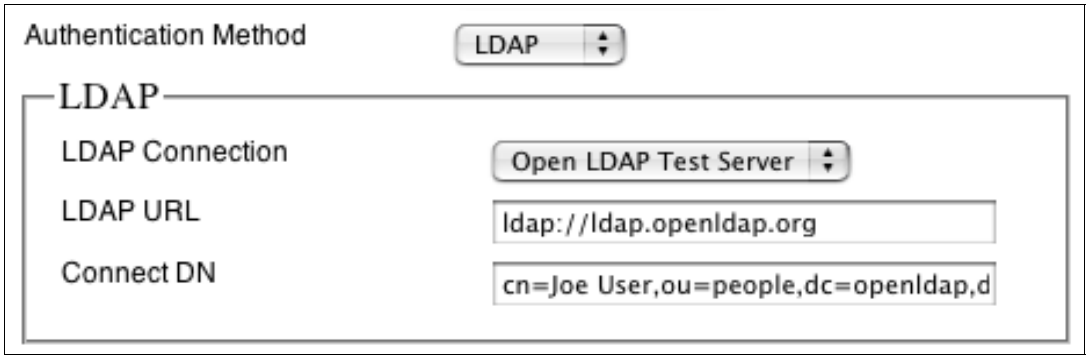
Anonymous Registration, when LDAP is the authentication method, means that users can register themselves with WebGUI as long as they exist in the LDAP directory. If they don't, they won't be allowed to register. As an admin this means you don't have to pre-create WebGUI accounts for all your users. They can do it for you.

Automatic LDAP Registration automatically creates a WebGUI account for

the user when s/he logs in using valid LDAP credentials. This allows LDAP users to skip the registration process entirely, making it easier for them to gain access to the site resources. As an admin, this gives you one more way to have the users create their own accounts so you don't have to

Are you wondering why you have to register your LDAP users with WebGUI? It's because WebGUI allows for many different types of authentication. In fact, each user can authenticate against an entirely different source. Therefore, you need to register your users so that WebGUI knows which source to use.

If you don't enable either of the above options, then you'll need to register each of your users manually with WebGUI. To do that, go to Admin Console > Users and click “Add a new user.”



For Authentication Method choose “LDAP”. Then choose the LDAP connection you created before for LDAP Connection. The LDAP URL should be fine as is. Then, under Connect DN you need to specify the fully distinguished name (DN) of this user.

Once you've done this, you can authenticate this user using his or her LDAP credentials.

Going forward, when a user updates his/her password in LDAP, those changes will automatically be reflected in WebGUI. In addition, if you disable the user in LDAP, the user will no longer be able to log in to WebGUI.

Group Inclusion

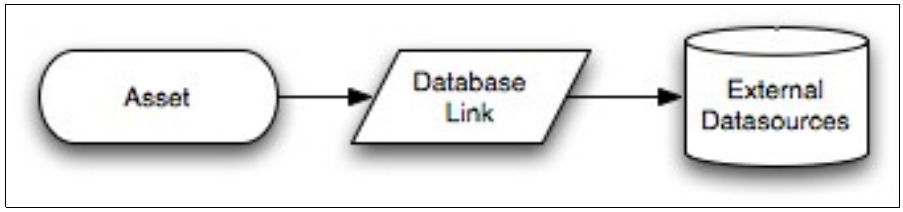
Now that you have your users authenticating using LDAP, you can also associate WebGUI groups to LDAP groups so that LDAP users that are in LDAP groups will automatically be included into the associated WebGUI Groups. To learn how to do this, see the chapter on WebGUI Groups, and specifically the section on Special Inclusion Via LDAP.

Database Links

To fully understand WebGUI database links you must first understand what they are used for. There are a couple of assets in WebGUI that have the ability to interact with data sources. Those sources could be the WebGUI database itself, another database on the same server that is hosting your website, a database that is running on a server half way around the world, or perhaps even a text file. There is a great deal of power when you have a system that can process data from multiple sources to generate useful information. In WebGUI, database links are one component of the system that makes this possible.

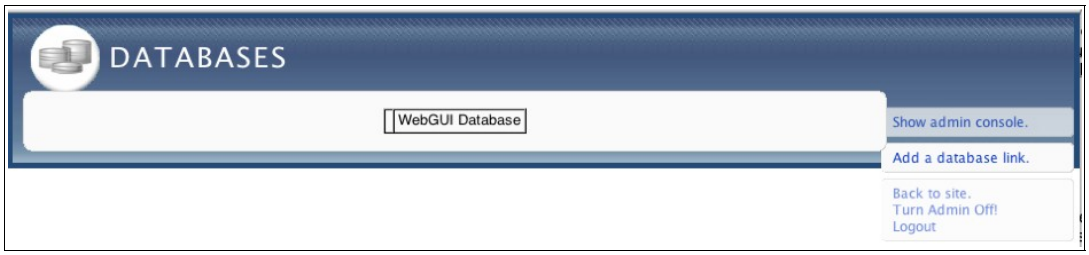
You can think of database links as a way for WebGUI to use data that is stored in other places. In order for WebGUI to be able to access the information, you have to tell it where it is and what credentials are needed to get to it. The function of a database link is to facilitate this connection and store the information needed to use it.

To gain a bit of perspective, let's look at the big picture. The way this all works together is that an asset in WebGUI with the ability to process external data will allow you to specify a database link for it to use in order to access that data. The database link can be used by multiple assets at the same time and only has to be defined once. The database link in turn handles all of the data manipulation the asset requested using a method that the data source you're using understands. The important thing to realize is that the asset using the link has absolutely no idea what type of database it is interacting with or what specific commands the platform requires. The database link facilitates this interaction in a generic way regardless of the data source.



There is one special database link with which you should be familiar. This is the default database link. The default database link is a link to the database that WebGUI uses in order to function. All of your users, groups, content, and many other things are stored there. You will see the default database link when you view the databases screen from the admin console. It is

labeled “WebGUI Database” and differs from other database links in that it can never be edited, copied, or deleted.

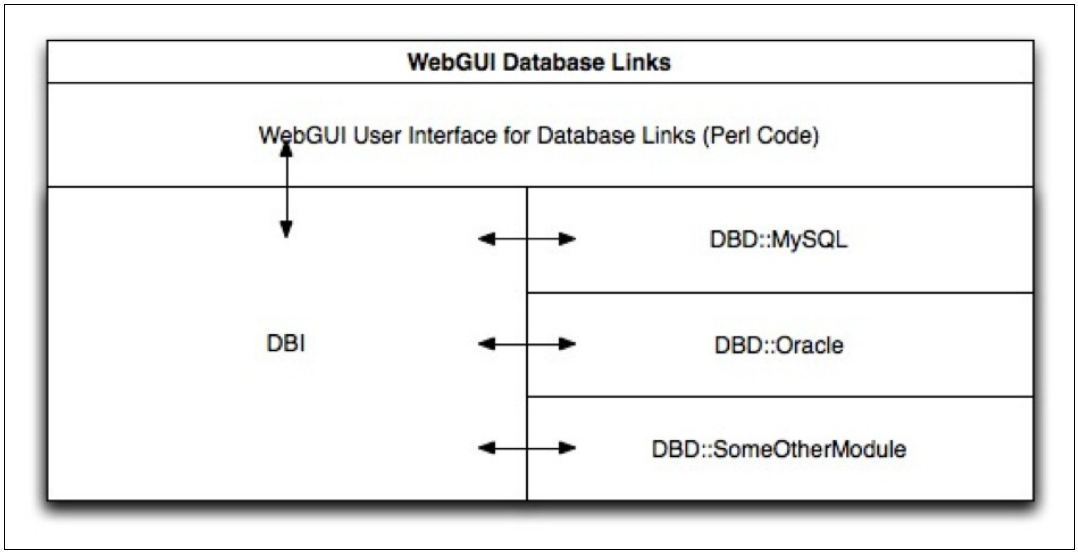


Because of this, those control icons are not listed next to the database link on the database page. Just as assets use a database link to communicate with data sources, so to does the core source code of WebGUI, and for the same reasons described in this chapter.

DBI and DSN

“Make everything as simple as possible, but not simpler” -- Albert Einstein

Einstein's quote is especially relevant to Perl DBI. Perl DBI is a programming library that allows code written in Perl to communicate in a consistent way with multiple data sources. With all of the various ways that exist for one to store data, it would be incredibly difficult, if not impossible, to keep up with them all, not to mention that the authors of Perl have absolutely no control over the rules that various database designers may implement for their systems. DBI was written to mitigate this problem by allowing perl programmers to use a standard set of commands to perform operations on a database, regardless of what type of database that is. To do this, DBI uses drivers for various data sources. You can find these drivers on the CPAN website by searching for “DBD”.



DBD stands for DataBase Driver and can be thought of as a translator that understands two languages. The first language it understands is DBI and the second language is the language that the database server uses. This could be MySQL, Sybase, Oracle, ODBC or any number of other data storage platforms that exist today. So, to expand on the earlier example of the "Big Picture", look a little deeper at the Database Link portion of the diagram.

A database link in WebGUI is really just a series of screens that allow you to interact with DBI and DBD in a user friendly way and save the settings you assign to each one.

It is important to remember that in order to connect to an external data source using a WebGUI Database Link, you must have the appropriate DBD installed for the database type you are trying to connect to.

Another thing you need to know about is called the DSN, or Data Source Name. This is nothing more than a specially formatted string of characters that tells DBD how to connect to the data source you want to connect to. The format the DSN takes is dependent on the DBD driver being used. Some examples of the more common DSN formats are available by using the WebGUI hover help feature by holding your mouse over the DSN property field in the WebGUI Database Link screen. An example DSN is also used later in this chapter for your reference.

An example transaction may go like this:

1. WebGUI asset code asks for DBI to create a connection to the

database defined in the database link that the asset is using. The database link the asset is using is selected by the content manager when the asset is added to the site.

2. DBI then looks to see which driver has been specified for the connection and gathers the location of the database server and the credentials necessary to authenticate to it. Additionally, it checks to see which database operations the WebGUI database link is configured to allow, such as reading, writing, updating, and deleting data from the database.

3. DBI connects to the data source, and the DBI commands sent by the asset are given to the DBD driver which translates them into commands that the database understands. The commands are then executed if they are allowed by the database link.

4. The database sends information back based on the command executed. This information is given to the DBD driver which translates it back into a DBI compatible format.

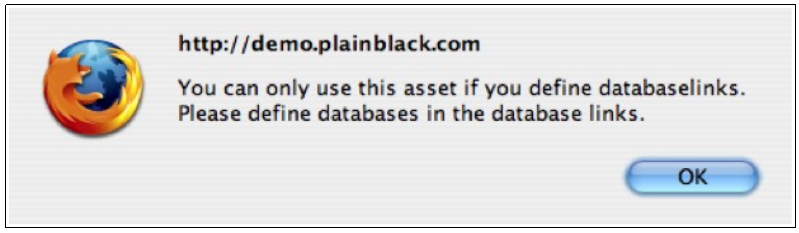
5. DBI hands the information off to the asset.

As you can see, the end result is that the asset code never has to change regardless of the location, format, or type of storage system being used. This allows WebGUI to access vast amounts of data in many different formats out of the box.

Database Link Usage

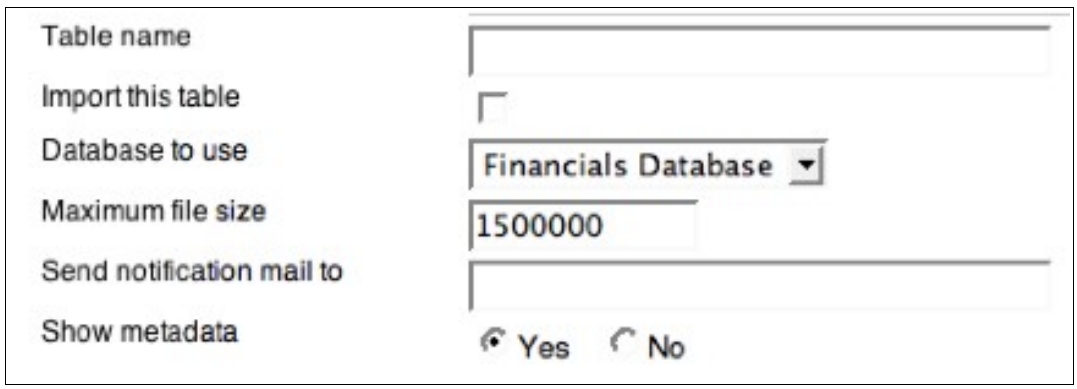
Now that you have a general idea of how a database link works, let's explore some of the places you can use one in WebGUI.

SQL Form Asset: The SQL Form asset allows you to interactively create the front-end for various database tables, much like you can with applications such as Microsoft Access. This asset needs you to specify the database link it should use to manipulate data in the table you specify. In fact, the SQL Form will not even allow you to add the asset to your site unless you have a database link defined. If you attempt to add an SQL Form asset without one defined, you will see an error message similar to this.



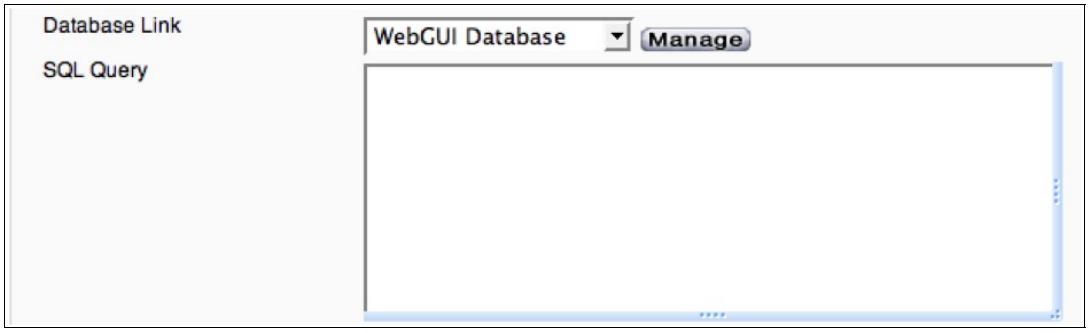
The SQL Form is unique in this regard because most assets will allow you to use the default database link. You will recall from the last section that the default database link is the one that allows you to work with the database used by your WebGUI installation. Because the SQL Form allows the user to create and alter data in the database it is important that this asset protect the WebGUI database by not allowing it to be manipulated directly from the user interface by a content manager or site visitor. There are many reasons for this and among them are the ever changing schema of the WebGUI database, the fact that altering certain information incorrectly could cause the entire website to stop functioning, and for security concerns. You wouldn't want a content manager to create a SQL Form on the groups table, for instance, because they could then circumvent all of WebGUI's security features and add themselves to whatever groups they please. There are endless other scenarios one could come up with that are equally terrifying for an administrator.

If, however, you have one or more database links defined, you can find the Database Link field for the SQL Form on the Properties page of the asset towards the bottom of the screen. The property is labeled "Database to use". All of the database links you have defined will be available in this drop down box, with the exception of the default database link.



SQL Report Asset: The SQL Report asset is used to pull data from a database for reporting purposes. The data is returned and processed through the WebGUI templating system so that reports can be presented however the content manager would like. In order for a report to be generated, however, the SQL Report needs to know which database link to use. Unlike the SQL Form asset, the SQL Report asset can access the default database link but only in a read-only fashion. The database link is set for an SQL Report on the properties tab of the asset at the bottom of the screen. By default, the default database link is selected.

WebGUI Groups: A WebGUI group can be configured to include users based on the result of a database query. More information on this feature can be found in the groups chapter. The database link property is defined on the edit group screen towards the bottom of the page.



See the Special Inclusion section of the Groups chapter for more information.

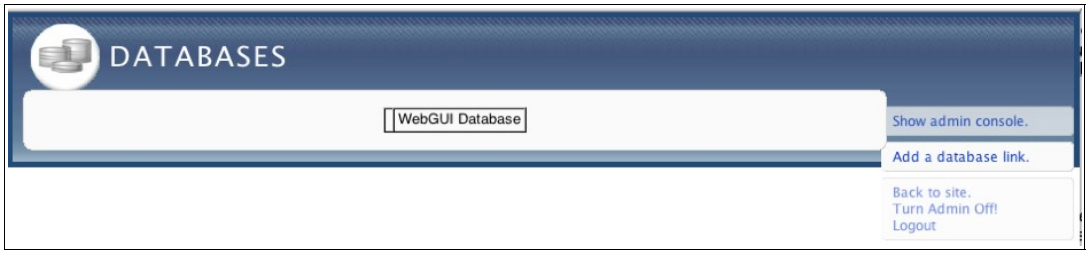
Custom Code: If a developer has written a plugin for WebGUI, such as an authentication module, asset, macro, etc. that you have installed on your site, it may have a database link property as well. Consult the documentation and help files supplied with the code for more information.

Configuring a Database

Now that you know where you can use a database link and how they work, let's talk about how you actually create one.

The first step is getting to the Databases management screen in the WebGUI admin console. This can be accomplished using the WebGUI Admin Console and clicking on the “Databases” icon.

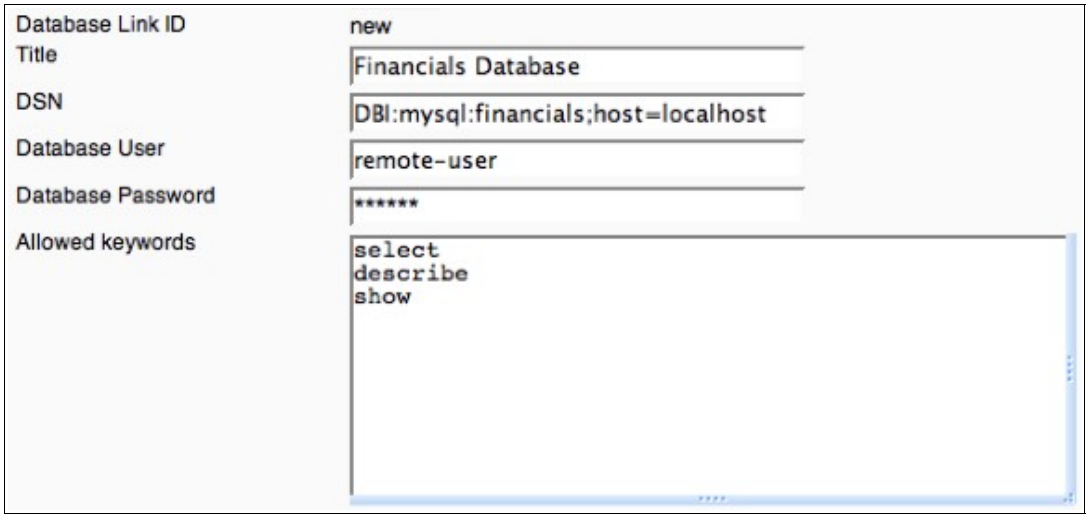
Clicking this icon will allow you to manage the database links for your WebGUI site. The default view lists all of the current links that are defined. Beside each link you will see three icons. The red “X” deletes the link, the “edit” icon allows you to modify the links properties, and the “copy” icon allows you to make a replica of the link with a new name. The copy functionality creates a copy of the connection information only, not the data source itself. This feature is useful if you are connecting to another database with very similar properties.



The default database link is listed in the center of the screen. On the right hand column you will see links to “Show Admin Console”, “Add a database link”, “Back to site”, “Turn Admin Off”, and “Logout”. All of these function exactly as they do in other admin console screens. The one link you’re interested in is the “Add a database link.” option. This link allows you to create a link to another data source.

The “Add a database link” screen is the same one displayed when editing an existing database link. The difference is that the property fields are pre-filled when editing previously saved values.

This example connects to a MySQL database that exists on the same server as the WebGUI installation. The database name is “financials” and the mysql username and password with access rights to this database are “remote-user” and “money!” respectively. For this link, we only wish to allow assets to use the MySQL commands select, show, and describe (read only access). Let’s go through the fields necessary to complete this example.



- Database LinkId: This field is auto-generated by WebGUI. When you first start editing it will be “New”, and after saving the details WebGUI will assign a unique id to the link. This is a read only field that you don’t need to worry about.
- Title: Specify any name you want here that is descriptive of the connection to be made. This is the name that your content managers will see when selecting a database link from the drop down list using the asset property screen.
- DSN: This is the Data Source Name property and specifies which DBD driver to use, the name of the database to use and various other properties depending on the DBD driver.

Generally the format is:

DBI:driver:database-name;property=value;property=value.

In this example you are connecting to a MySQL database called financials that exists on the local server so your DSN is:

DBI:mysql:financials;host=localhost

- Database User: This is the user name that is sent to authenticate to the database server when attempting to make a connection.
- Database Password: This is the password that is sent to authenticate to the database server when attempting to make a connection.

- Allowed Keywords: These are the commands this database will allow to be sent to the database server. This feature will allow you to limit the set of commands that a database user can execute, even if the user has privileges to do so at the database level. This feature will not allow you to execute any commands to which the authenticating user does not have permission to do at the database level.

Now, when you click the Save button, you return to the manage database links page and your new database link is there.

Scratch Variables

Scratch variables are usually used by programmers to store small bits of information about a user or a user's preferences for the duration of their session in WebGUI. However, scratch variables can also be set via the URL. The latter is what this chapter is all about.

Setting

To set a scratch variable, you just add a few fields to the end of the URL in your site. For this example, assume that your site is example.com.

```
http://example.com/?op=setScratch;scratchName=X;scratchValue=Y
```

Take a closer look at the three parameters.

```
op=setScratch
```

This tells WebGUI you want to set a scratch variable.

```
scratchName=X
```

This tells WebGUI you want to set the value of a scratch variable called

```
www_X
```

Any scratch variable set through the URL will be prefixed with www_. This is done automatically so don't include it with your variable name. It is done this way as a security precaution so that users are not able to maliciously change how a program works without the programmer knowing about it.

The value of www_X is then set with the following:

```
scratchValue=Y
```

This tells WebGUI the value of **www_X** should be **Y**.

Deleting

To delete a scratch variable, you just add some fields to the end of your site URL like this:

```
http://example.com/?op=deleteScratch;scratchName=X
```

Take a closer look at the parameters.

```
op=deleteScratch
```

This tells WebGUI you wish to delete a scratch variable.

```
scratchName=X
```

This tells WebGUI that the variable you would like to delete is called

```
www_X
```

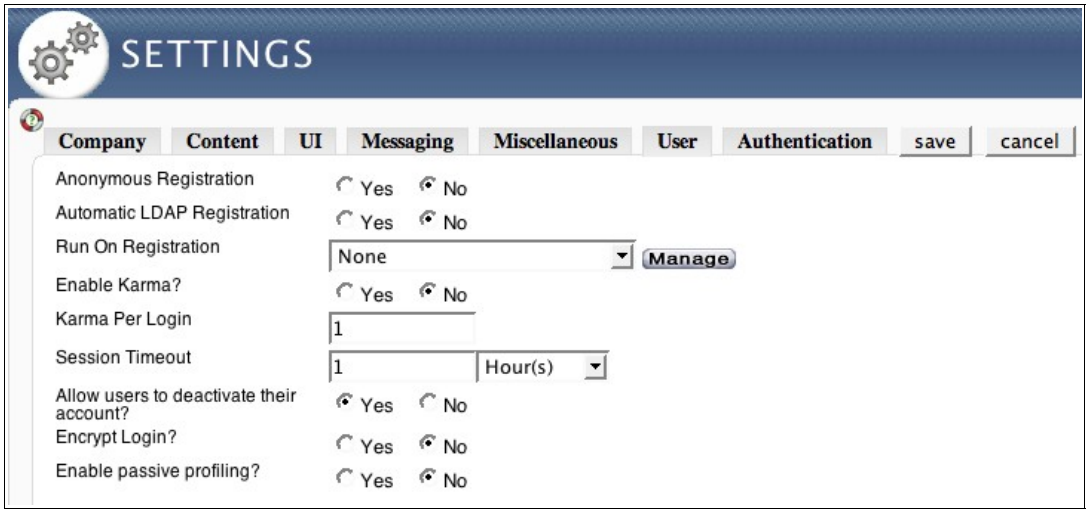
Uses

You may want to display something in a scratch variable by creating a custom macro, or you may want to use these in your own custom assets. The most popular use, however, is using scratch variables for special inclusion in groups. See Scratch Variables in the “Groups” chapter for details. In this case, you’re dynamically adding a user to a group for the duration of the session because s/he has clicked on the link you have provided. This is most commonly used for legal pages where you want the user to agree to something before you give them access to it.

Karma

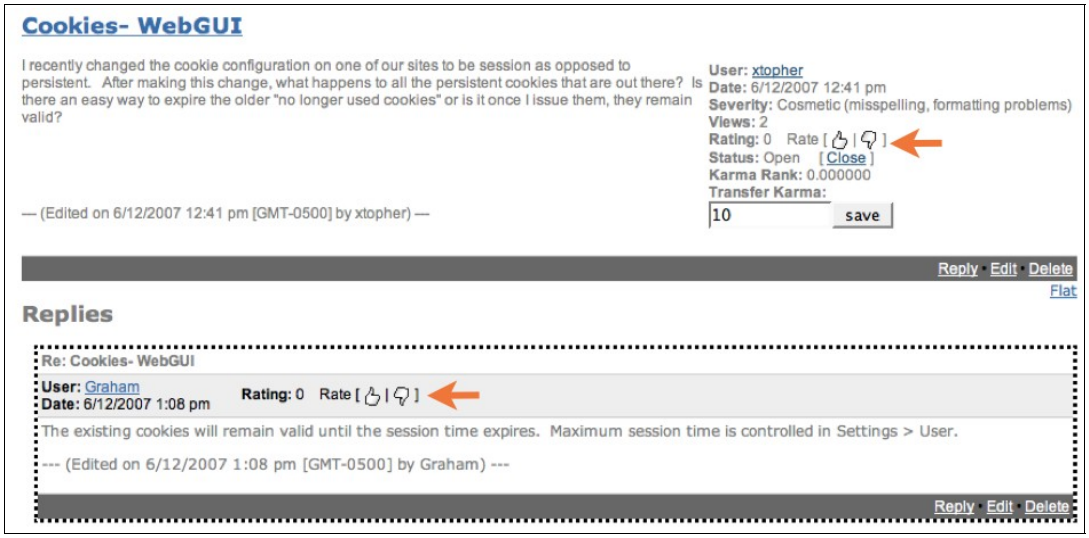
Plain Black developed karma as a sort of reward system for community participation. When WebGUI community members participate in a forum, vote in polls, or in other ways interact with the community users' karma increases. Karma can also be gained through the purchase of goods and services. Users can then turn around and “spend” their karma in a variety of ways throughout the site. For instance, karma can be spent on a request for enhancement to increase the odds of a new feature being implemented in WebGUI; karma can also be transferred to and from other users by rating forum posts up or down.

Karma is enabled in the User tab of the Settings screen in the Admin Console.



To enable karma on your site, simply set the “Enable Karma?” field to Yes. Users can also be granted karma for simply logging in to the site. In the “Karma Per Login” field you can designate the amount of karma gained each time a user logs in.

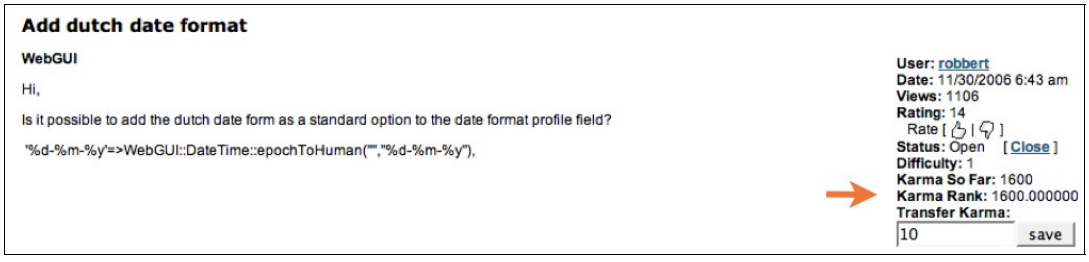
When karma has been enabled on the site, you will notice that some assets contain new fields. Most often this is visible in a Collaboration System. For example, a simple forum will display an area in which posts may be rated up or down.



When a user clicks the rating field up or down, karma will be transferred or deducted between users accordingly. So, by rating a post up, you would transfer a small amount of karma from your user account to the user whose post you have ranked. The appearance of the rating field may differ depending on the template used.

In a collaboration system set up as a request tracker a field will become available in which users may “spend” a specified amount of karma. On WebGUI's site this is used to move a request for enhancement up the list to ensure that it is more quickly developed.

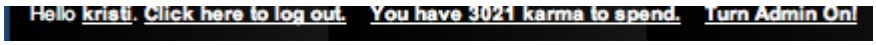
You can't rate your own posts, so the rating field will not be visible in the post you have made. However, you will be able to see what others have rated it.

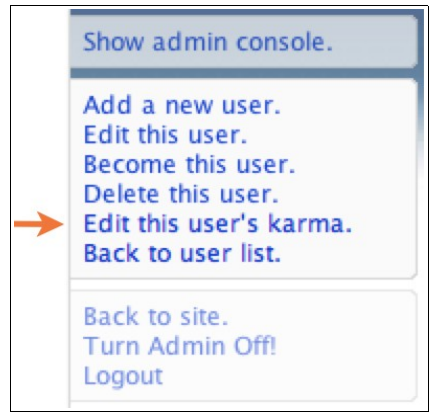


In this example, you can see three new karma related fields have appeared. “Karma so far” and “Karma Rank” indicate the amount of karma spent on this request so far. Near the bottom you see a “Transfer Karma” field. In this field a user can input an amount of karma to be transferred from the user account to this request. The default is 10. Once a value has been entered, the user simply clicks Save to “spend” the defined amount of karma on this

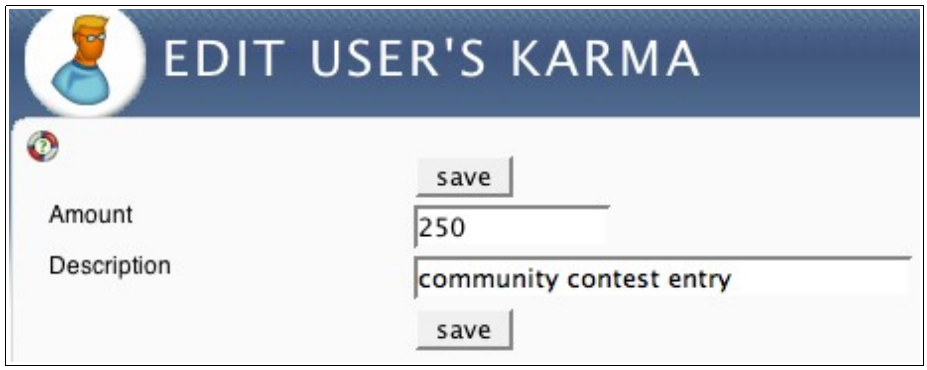
request.

You can also display to a user how much karma the user currently holds by using the Karma Macro. This will display the karma a user holds much like what you see on www.webgui.org.





A user's individual karma can be managed through the User screen of the Admin Console. In the Users screen, search for the specific user whose karma you would like to manage. In the user's account screen you will see an “Edit this user's karma” link on the far right hand side. Clicking on this link will open the “Edit User's Karma” screen. On this screen an amount of karma can be added by entering a positive amount in the “Amount” field, and karma can be deducted by entering a negative amount. In the “Description” field you may enter a brief explanation for the karma alteration. Upon clicking save, the user's karma will be appropriately modified to indicate the changes you have made.



Content Filters

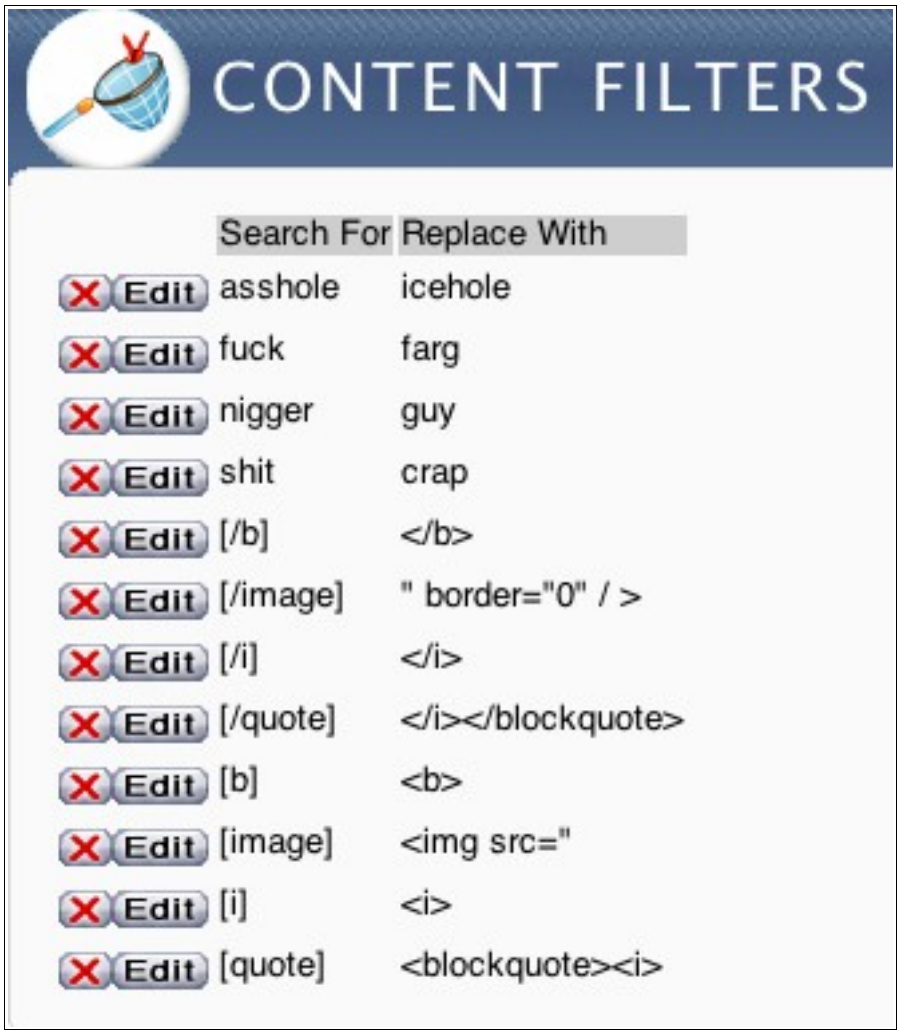
Content filters (also known as “Replacements”) are WebGUI’s way of allowing administrators to have some automated control over user submitted content. They can be used to give some functionality to a user, but more often than not they are used to curb profanity.

Note that we use many words in this chapter that some people consider vulgar. Please understand that this is not done to be inconsiderate, but just to provide examples of the most common use of Content Filters.

Where are Filters Used

As of this writing, content filters are used in the Collaboration System asset and the Wiki asset. However, there is an API provided to the replacements system so other custom assets may use them as well.

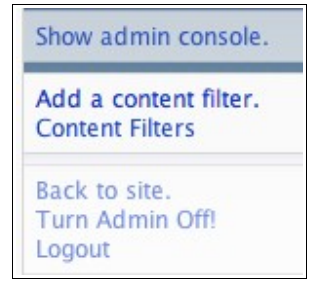
To access the Content Filter system, click on Content Filters in the admin console. This will open the Content Filters screen, on which all current terms and replacements will be displayed. There are some defaults that come set up with the system.



Adding A Filter

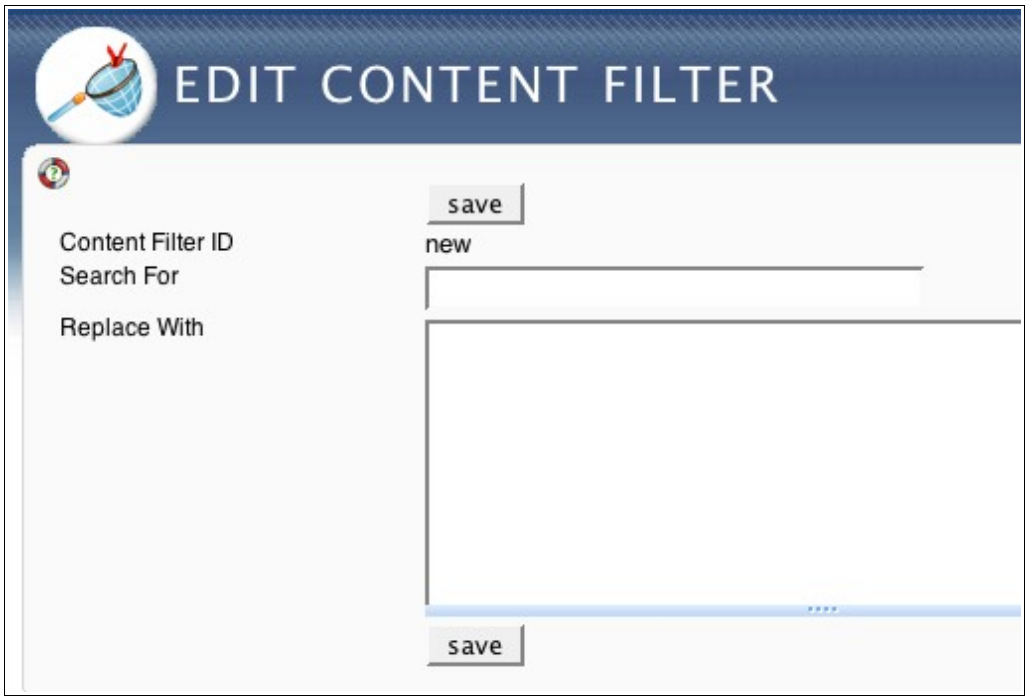
Adding and editing content filters is easy, just follow these instructions:

1. Log in and go into admin mode.
2. Select “Content Filters” from the admin console.
3. You will be presented with a list of content filters that have already been created. The system comes with about a dozen of the common ones.



4. From here, click on the “Add a content filter” link on the right side of the screen.

5. You’ll be presented with a short form.



6. The first field is “Search For”. In this field you enter something you want WebGUI to look for in users' posts. Let’s assume that you never want someone to use the word “bitch” on your site, because your site has nothing to do with dogs, and therefore “bitch” would only be used in a hurtful context. So enter “bitch” in the “Search For” field without the quotes.

7. The second field is “Replace With”, which is what you want WebGUI to output in place of the original item. So in this case, let’s output some Q-Bert™ language. Enter “@#$!” into the “Replace With” field, again without the quotes.

8. Click “save”.

You’ve successfully created a content filter that can be used in Wikis and Collaboration Systems that have replacements enabled.

Recipes For Foul Language

It's easy to filter out provocative and vulgar language simply by putting the most common hurtful words into replacements.

However, often the people using such language are teenagers and angry adults, both of whom are devious human beings. If they know you filter the words then they will start using other replacements for common characters and symbols. You'll need to catch those as well. Here are some common ones you should look out for:

@$$ = ass

(o)(o) = boobs

f*ck = fuck

The good news is that content filters are retroactive, so if people come up with new terms you don't want to see on your site, you can always create a new content filter and it will automatically apply itself everywhere that content filters are enabled. In addition, people trying to circumvent your filters will often times make themselves look unintelligent simply because their language will be changed by your filters.

Exposing Functionality

Often times you'll set up your rich editors and your other filters on your Collaboration Systems and Wikis to be very strict, not allowing much in the way of HTML to get into posts. However, you can use replacements to expose that functionality to your users. For example, you could do something like this to allow your users to format source code:

Search For	Replace With
[code]	<blockquote style="border: 1px solid black; font-family: courier;>
[/code]	</blockquote>

Then your users can create a block of content like this:

```
[code]

my @colors = qw(black blue orange);

foreach my $color (@colors) {

        print “$color\n”;}[/code]
```

Which will be formatted like this:

```
my @colors = qw(black blue orange);
foreach my $color (@colors) {
     print “$color\n”;
}
```

Another example is to expose common symbols this way.

Search For	Replace With	Creates
[TM]	&trad;	™
(C)	©	©
webgui	WebGUI®	WebGUI®

Notice the last example? If you have a particular trademark or product name or something like that, and you always want it to appear a certain way, then creating a replacement for it might be just the trick.

Importing Files

Sometimes content managers will have more files to import than you'd probably want to import via a web interface. WebGUI comes with a tool to help you out. If you type the following commands:

```
cd /data/WebGUI/sbin
perl fileImport.pl --help
```

You will see the following output:

```
Usage: perl fileImport.pl --pathToFiles=<pathToImportFiles> --configfile=<webguiConfig>
--parentAssetId=<assetId>

    --configFile        WebGUI config file.

    --pathToFiles       Folder containing files to import.

    --parentAssetId     The asset ID of the asset you wish
                        to attach these files to.


Options:

    --groupToEdit       The group ID of the group that should
                        have the privileges to edit these
                        files. Defaults to '4' (Content Managers).

    --groupToView       The group ID of the group that should
                        have the privileges to view these
                        files. Defaults to '7' (Everybody).

    --help              Display this help message and exit.

    --owner             The user ID of the user that should
                        have the privileges to modify these
                        files. Defaults to '3' (Admin).

    --override          This utility is designed to be run as
                        a privileged user on Linux style systems.
                        If you wish to run this utility without
                        being the super user, then use this flag,
                        but note that it may not work as
                        intended.
```

```
    --quiet             Disable output unless there's an error.

    --webUser           The user that your web server runs as.
                        Defaults to 'apache'.

    --skipOlderThan     An interval defined in second to skip file older than.
                        Defaults "nothing skip".

    --findByExt         Import only files files with an extension matching
                        one of the exensions.
                        Defaults "import all files".

    --recursive         Import the files recursivelly from the folder --pathToFiles
                        Defaults "don't run recursivelly"

    --overwrite         Overwrite any matching file URL with the new file rather
                        than creating a new Asset for the file.
                        Instanciate the existing asset and replace the file.

    --ignoreExtInName   Title and menuTitle database fields should not contain the
                        extension of the filename.


EXIT STATUS

 The following exit values are returned:

 0
    Successful execution.

 1
    For Windows User, stop the script if not super user.

 2
    A folder can't be open for reading.

 3
    In recursive mode, if two files has the same name and are selected to be imported.
Return this error.

 4
    Error during invocation of the command.

 5
```

```
    The parent Asset Id doesn't exist.
```

Using this information you can quickly and easily import a whole folder full of files . Assuming you have a bunch of files at /home/me/files and you have an asset (usually a folder) with an assetId of “39z939sjjxhhHXl-LJnn1”, then you can bring all those files into WebGUI with a simple command like this:

```
cd /data/WebGUI/sbin
perl fileImport.pl --pathToFiles=/home/me/files –config=www.example.com.conf –
parentAssetId=39z939sjjxhhHXl-LJnn1
```

If the folder contained a whole bunch of files and you only wanted to import the JPEG's then you could add this to the end of that command:

```
--findByExt=jpg
```

And if your folder had lots of subfolders of files you could append this to the command to have it import everything under all the folders:

```
--recursive
```

As you can see, it's very easy to quickly import a whole bunch of files into your WebGUI site.

If you've already imported a bunch of images into your site, but you realize now that you want the thumbnails to be larger or smaller, WebGUI provides a tool for that too. Type this command:

```
cd /data/WebGUI/sbin
perl thumbnailer.pl --help
```

It will give you this output:

```
Usage: perl thumbnailer.pl --path=/path/to/files [--size=thumbnailSize] [--missing]

--path is the complete path to your uploads directory

--size=thumbSize allows you to override the default thumbnail size of 50.

--missing says to only create thumbnails for images that are missing thumbnails.
```

Therefore, by using the following command you can resize all your thumbnails site-wide to 100 pixels:

```
cd /data/WebGUI/sbin
perl thumbnailer.pl --path=/data/domains/www.example.com/public/uploads --size=100
```

Exporting Content

When you have a system as powerful as WebGUI it seems unlikely that you'd ever want to export your content out of it. However, there are lots of reasons that you may wish to do just that:

- You need some content that can be easily formatted into an email or some other external publication.
- Your web host only allows HTML uploads, but you still want to use WebGUI to create the content.
- You need to run the content on multiple caching servers around the globe.
- Certain pages in your site are hit very hard all the time, so it's a better use of system resources to make those pages as static HTML.

Whatever your reason for wanting to export your content, WebGUI provides three mechanisms to help you do it: Single Page, Branch, and With Commit

Exporting a Single Page

Exporting a single page in WebGUI is quite trivial. Run the following command:

```
cd /data/WebGUI/sbin
perl generateContent.pl --help
```

That will produce the following output:

```
Usage: perl generateContent.pl --configFile=<webguiConfig> --url=home

Options:

	--configFile	WebGUI config file (with no path info).


	--assetId	Set the asset to be generated.

	--help		Displays this message.

	--styleId	Set an alternate style for the page.
```

```
                Defaults to asset's default style.

    --toFile      Set the path and filename to write the
                content to instead of standard out.

    --url         The URL of the asset to be generated.

    --userId      Set the user that should view the page.
                Defaults to "1" (Visitor).
```

Therefore, you could export the home page like this:

```
cd /data/WebGUI/sbin
perl generateContent.pl --conf=www.example.com.conf --url=home
```

And you could export it to a file called “index.html” by appending the following onto the command:

```
--toFile=index.html
```

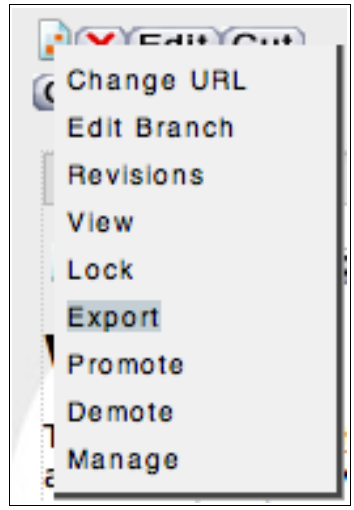
This can be useful if you only need to export a page or two, or if you want to export content in a scripted fashion, such as running it from a cron job.

Exporting a Branch

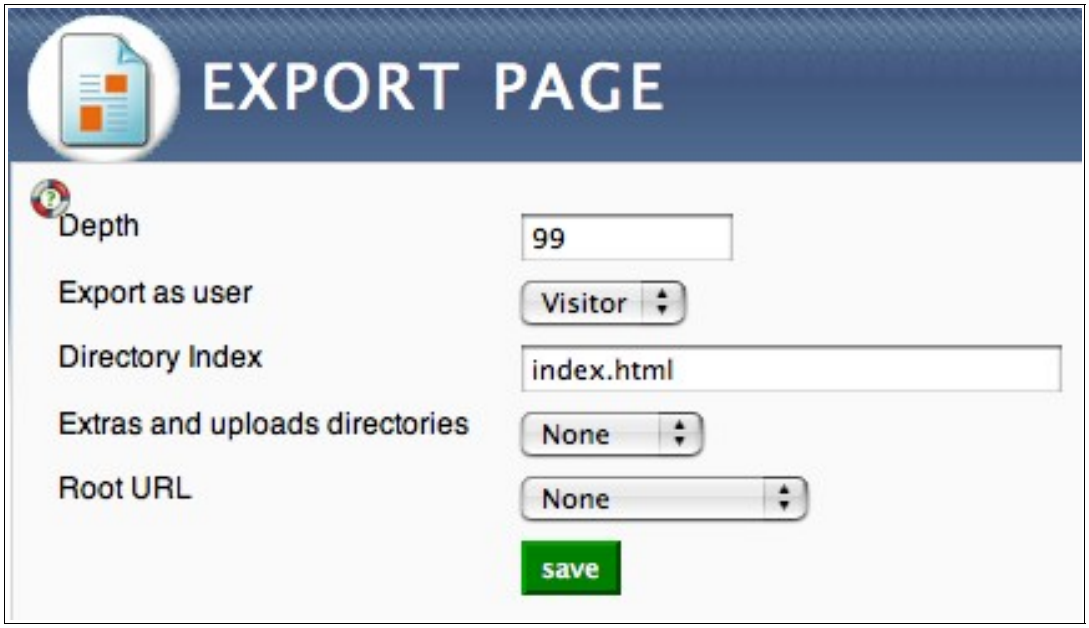
WebGUI also has a mechanism to allow you to export an entire branch of content whenever you like. To use this you first must enable it in your WebGUI config file. You need to add a directive like this:

```
“exportPath” : “/var/www/exports”,
```

Then restart modperl to have the change take effect. Going forward, anyone in the “Export Managers” group will be able to export a branch of content. The export function is available under the class icon context menu of all assets.



That will bring you to the export page. Here you can choose how many levels deep in the branch you'd like to export, what user should be logged in for exporting the pages, the default file name your web server will look for on directory index files, and finally whether or not to create symbolic links for extras, uploads, and root urls.



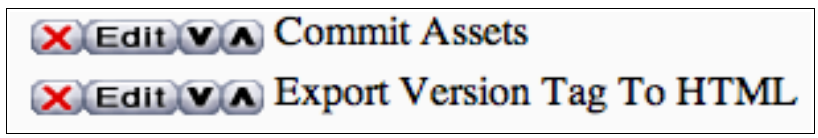
When you hit “save”, you'll be presented with a screen that shows you the export status in real time:

```
Exporting page /tmp/export/your_next_step/index.html .....DONE
Exporting page /tmp/export/yns/translated/index.html .....DONE
Exporting page /tmp/export/yns/support/index.html .....DONE
Exporting page /tmp/export/yns/style/index.html .....DONE
Exporting page /tmp/export/yns/promotion/index.html .....DONE
Exporting page /tmp/export/yns/hosting/index.html .....DONE
Exporting page /tmp/export/yns/features/index.html .....DONE
Exporting page /tmp/export/yns/experts/index.html .....DONE
Exporting page /tmp/export/yns/docs/index.html .....DONE
Exporting page /tmp/export/the_latest_news/the_latest_news/index.html .....DONE
Exporting page /tmp/export/the_latest_news/index.html .....DONE
Exporting page /tmp/export/tell_a_friend/tell_a_friend/index.html .....DONE
Exporting page /tmp/export/tell_a_friend/index.html .....DONE
Exporting page /tmp/export/site_map/site_map/index.html .....DONE
Exporting page /tmp/export/site_map/index.html .....DONE
Exporting page /tmp/export/home/welcome/index.html .....DONE
Exporting page /tmp/export/home/key-benefits/index.html .....DONE
Exporting page /tmp/export/home/ad2/index.html .....DONE
Exporting page /tmp/export/home/ad/index.html .....DONE
Exporting page /tmp/export/home/index.html .....DONE
Exporting page /tmp/export/getting_started/getting-started-part2/index.html .....DONE
Exporting page /tmp/export/getting_started/getting-started/index.html .....DONE
Exporting page /tmp/export/getting_started/index.html .....DONE

Exported 23 pages in 5 seconds.

Back to site.
```

Exporting With Commit

You can also export content as you make modifications to it. In this way you don't have to export content that you've already exported before, and you don't have to remember to export it. Note that this option uses the “exportPath” config file directive too. See the previous section for details.

To set this up, go to Admin Console > Workflow and edit your content approval workflow. In this example, the “Commit Without Approval” workflow is edited. Then, add the “Export Version Tag To HTML” activity so that it has two activities, like this. Be sure the export happens after the commit, or you won't get the results you expect.



From now on when you commit your changes, those changes will also be exported out to static HTML in the export folder defined in your config file.

Exporting Caveats

There are a few things to be concerned about when exporting content from WebGUI.

1. WebGUI has a lot of dynamic features like polls, login macros, etc. that will not work in a statically exported site. Be sure not to use any of that stuff in your content if you plan to export.

2. If you don't put file extensions on your asset URL's, WebGUI will export them as a folder and then create an index.html inside that folder with the content from that asset. If you don't like the idea of those folders, then use file extensions on all your URL's.

3. If you upload your files to another server, you should also upload your WebGUI uploads folder, as it will contain images and files that your site likely needs.

Search Indexing

WebGUI has a pluggable search indexer, which allows you to index, and therefore search, not only your WebGUI content, but any files attached to the content.

The Indexer

Before getting into extending the abilities of the indexer, we should first talk about how to index content. An indexer is a program that catalogs content into keywords and phrases so that it can be rapidly searched. WebGUI automatically indexes your content as you commit your changes. However, you can also tell WebGUI to manually re-index your content. To do this, use the command line utility search.pl found in your WebGUI/sbin folder. If you run the following command you'll see what options are available:

```
cd /data/WebGUI/sbin
perl search.pl --help
```

```
perl search.pl [ options ]

Options:

    --configFile=       The config file of the site you wish to perform
                    an action on.

    --help              Displays this message.

    --indexall          Reindexes all the sites. Note that this can take
                    many hours and will affect the performance of the
                    server during the indexing process.

    --indexsite *       Reindexes the entire site. Note that depending
                    upon the amount of content you have, it may take
                    hours to index a site and server performance will
                    suffer somewhat during the indexing process.

    --search= *         Searches the site for a keyword or phrase and
                    returns the results.

    --updatesite *      Indexes content that has not be indexed, but does not
                    index content that has been indexed. This is useful
                    if the --indexsite option had to be stopped part way
```

```
                through.

    * This option requires the --configFile option.
```

Why would you want to manually re-index your content? There are a number of reasons:

- You just added some plugins for additional file types and you want WebGUI to re-index the content so that it can index all your existing assets with those file types.
- There was a change to the search system, or there was a bug, and the WebGUI gotcha.txt file tells you to re-index your content.
- You performed some manual changes to the database, either through external content imports, site splits or merges, or some other external function and you want to make sure that all the changes are indexed correctly.

To reindex the content on your site, simply type:

```
cd /data/WebGUI/sbin
perl search.pl --config=www.example.com.conf --indexsite
```

You can also search your content from the command line to make sure that the indexing worked as you expected it to. To do that, type:

```
cd /data/WebGUI/sbin
perl search.pl --config=www.example.com.conf --search=features
```

The word “features” in the above command was the keyword searched for. If the search found nothing it would output something similar to the following:

```
Search took 0.048402 seconds.
```

If it did find some content, then the output would look like this:

```
4Yfz9hqBqM8OYMGuQK8oLw  Get Features
OhdaFLE7sXOzo_SIP2ZUgA  Welcome

Search took 0.025347 seconds.
```

The first column is the asset id of the asset it found, and the second column is the title of the asset.

Adding File Types

The WebGUI search indexer allows you to catalog attachments in addition to your WebGUI content. This is done through the use of command line programs that can turn file content into either text or HTML. The program must return the content to standard out. A good example of this is the “cat” program that comes with all Unix®, Linux®, and BSD systems. If you type

```
cat /path/to/product.html
```

it would output the contents of the product.html file. In the case of files that contain only HTML or text, the cat program is a perfect way to index those files. Unfortunately, it's not so easy for most binary application files, like those created by office productivity software.

Luckily, the WRE comes with a couple of utilities that will convert Microsoft® Word (catdoc) and Adobe® PDF (xpdf) files into text. And when you create your site using the WRE, it automatically adds them to your WebGUI config file.

On a Unix®, Linux®, or BSD style operating system a section like this will be added to your WebGUI config file.

```
  "searchIndexerPlugins" : {
     "doc" : "/data/wre/bin/doc2txt.sh",
     "pdf" : "/data/wre/bin/pdf2txt.sh",
     "readme" : "/bin/cat",
     "txt" : "/bin/cat",
     "html" : "/bin/cat",
     "xls" : "/data/wre/bin/xls2txt.sh",
     "htm" : "/bin/cat",
     "ppt" : "/data/wre/bin/ppt2txt.sh",
     "rtf" : "/data/wre/bin/doc2txt.sh"
  },
```

Unfortunately, not all of the same utilities are available for Windows® users. Your config file will have to be modified to look like this:

```
   "searchIndexerPlugins" : {
     "doc" : "/data/wre/bin/doc2txt.bat",
     "pdf" : "/data/wre/bin/pdf2txt.bat",
     "readme" : "/data/wre/bin/cat.bat",
     "txt" : "/data/wre/bin/cat.bat",
     "html" : "/data/wre/bin/cat.bat",
     "htm" : "/data/wre/bin/cat.bat",
     "rtf" : "/data/wre/bin/doc2txt.bat"
  },
```

If you can find, buy, or build other utilities to convert other document types. You can add them to your config file. The first parameter is the file extension to look for, and the second parameter is the path to the program that will convert that file into text or HTML.

Note that as you add new programs to your search indexer plugins, they will not retroactively index content that is already on your site. For that you need to re-run the indexer previously described. However, those plugins will be used for any new content added to your site.

Attachment Type Icons

If you've ever uploaded a file to WebGUI, you've probably noticed that it automatically displays an icon representing the type of document you've uploaded. The icons are always the icon you'd see in Windows Explorer or Mac Finder when looking at files on those operating systems.

Unfortunately, there is no way for WebGUI to automatically extract the icons from the files that are uploaded, so it has to do a little bit of magic to fool you into thinking that's what it has done.

In your WebGUI installation, take a look at the contents of this folder:

```
/data/WebGUI/www/extras/fileIcons
```

You'll notice there are a few dozen small image files in that folder. Each file is an icon for a file type.

Adding New File Types

Why would there be a chapter in this book about attachment type icons unless you could modify them! The good news here is that WebGUI's internal code knows nothing about these icons except for the folder they are in. Therefore, you can add new types. For example, you may have a CAD system in-house that's used to create drawings, and then upload those drawings to your intranet that's built on WebGUI. WebGUI has no way of knowing what every program out there is, so this enables you to add your own icons for the programs that your organization uses in-house. Here's how:

1. Create an image file that is 16 pixels wide, and 16 pixels tall, which includes the icon for the document type you'd like to add.

2. Save that file as a GIF with a filename of ***extension***.*gif* where the word extension is replaced with the document's normal file extension. For example, a Microsoft Word® document has an extension of ***doc***.

3. Place the image into */data/WebGUI/www/extras/fileIcons*.

That's all there is to it. Any files you've already uploaded with that extension will now start using that icon. Likewise, any files you upload in the future that use that extension will also use that icon.

You can add as many icons for as many programs as you like in this way. If it's a common program and you think that its icon should be added to the core then you can file an RFE for it. See the "More Resources" chapter later in this book on how to submit an RFE.

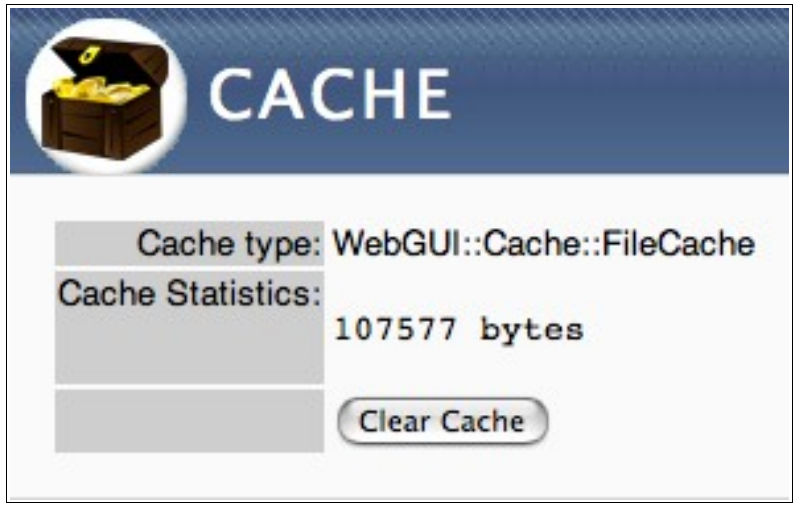
Clearing WebGUI Cache

In this book, and in various utilities within WebGUI, you'll be asked to clear your cache once in a while. That's usually because your cache has been poisoned by some manual change that you've just done. A poison cache can wreak havoc on WebGUI, and can even cause database corruption, so when you're asked to clear the cache, make sure you do it!

Cache can be cleared in one of two ways. The fastest way to clear the cache is manually from the command line. The easiest way is to do it through the WebGUI user interface. We'll show you both.

The User Interface

To clear the cache through the WebGUI user interface go to Admin Console > Cache. You'll be presented with a screen that looks like this:



To clear the cache, simply click on the “Clear Cache” button.

You might say to yourself, “That was both easy and fast.” You'd be right in most cases. If you're using the FileCache backend and you have a large site, you might find that it takes a really long time for this screen to be displayed, or it may never show up. That means you've outgrown the FileCache backend, and need to upgrade to the Database cache backend. See the WebGUI Config File chapter for how to change cache backends.

From the Command Line

You can also clear the WebGUI cache from the command line. However, to do this you need to know which type of cache backend your site(s) is using. Note that each site might be using something different. Look in your config file for the “cacheType” directive to find out which kind of cache each site is using.

FileCache

To clear the FileCache backend you should run this command:

```
rm -Rf /tmp/WebGUICache
```

On Microsoft® Windows® you'll run this command:

```
del /s /q c:\temp\WebGUICache
```

Note that the FileCache module allows you to specify a new cache location. Look for “fileCacheRoot” in your config file to see if it has been altered.

If you want to delete the cache just for a specific site, then you'd want to do the same command, but append the config file name to the end of it like this:

```
rm -Rf /tmp/WebGUICache/www.example.com.conf
```

Database

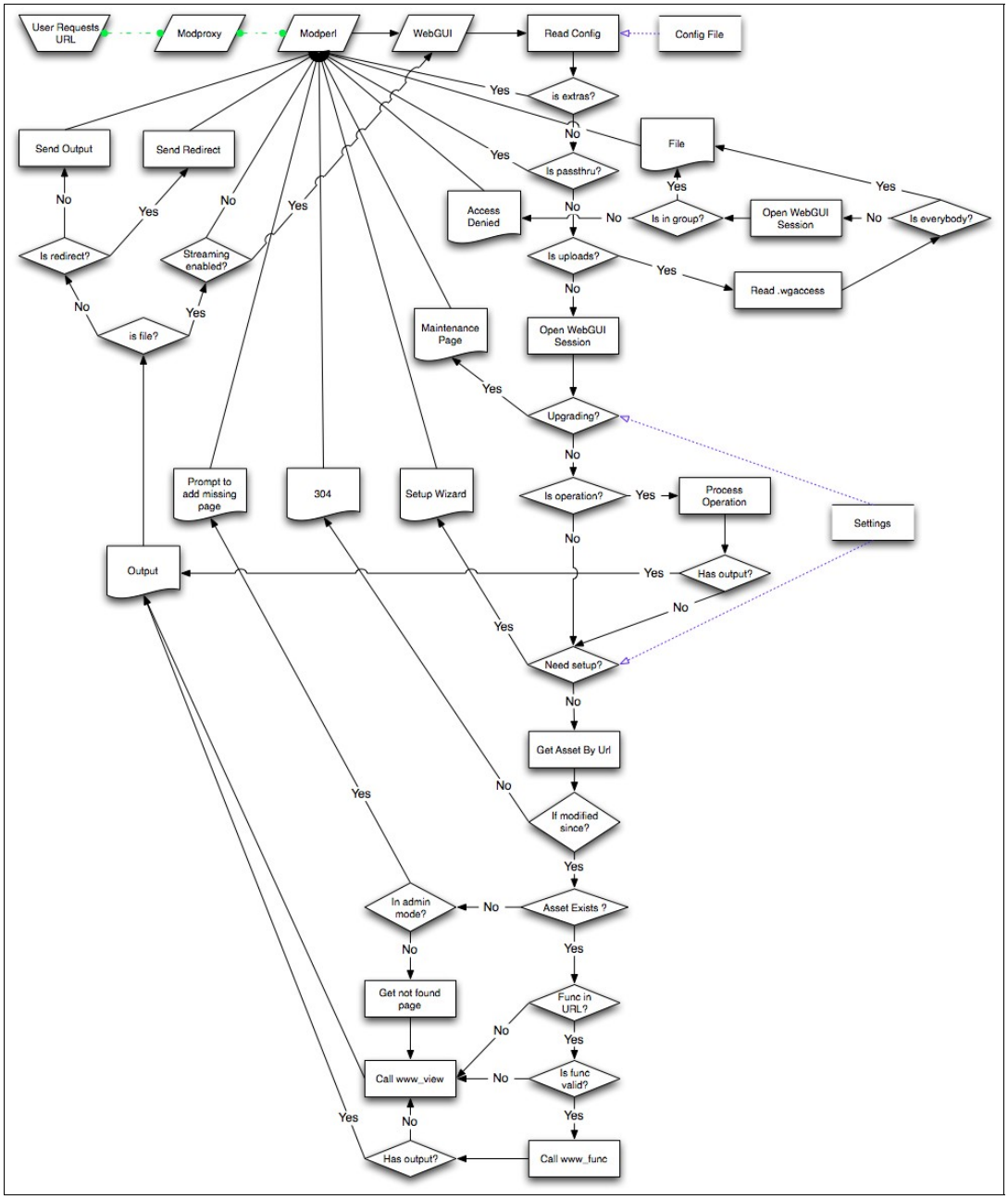
If you're using the Database cache backend then you can clear your cache with these commands:

```
mysql -uUSER -pPASS www_example_com
delete from cache;
exit
```

Be sure to get your username, password, and database name from your WebGUI config file.

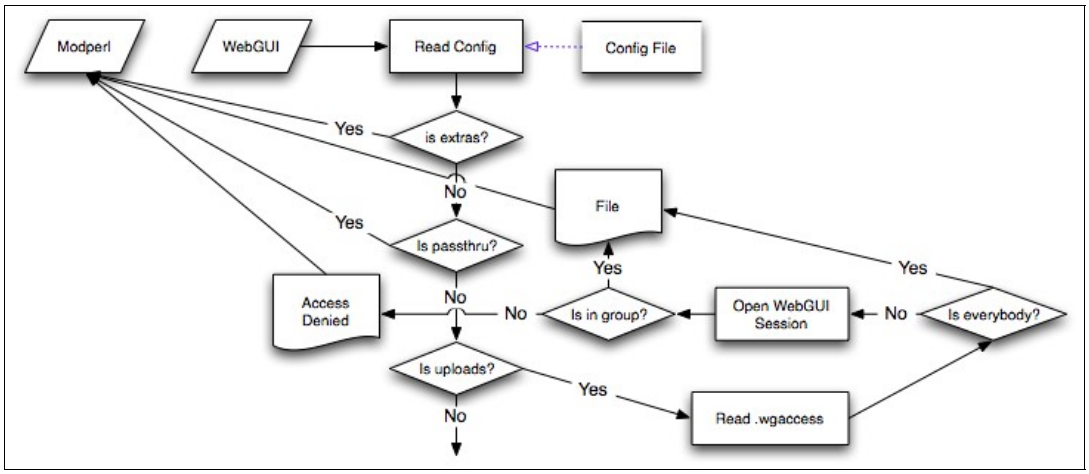
WebGUI Request Cycle

This chapter will give you an idea of the inner workings of WebGUI. It may help you troubleshoot performance problems or errors that you're seeing. Here's a high level diagram of the request cycle. Don't be afraid. It will be covered in a little more detail throughout this chapter.

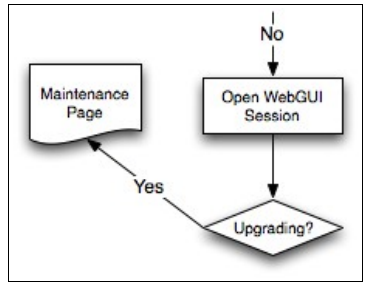


To begin, a user via its web browser, or someexternal service via some code, will request a WebGUI page. It first hits Apache mod_proxy, which then forwards it on to Apache mod_perl. Since WebGUI is registered as a handler for mod_perl, mod_perl then passes the request on to WebGUI.

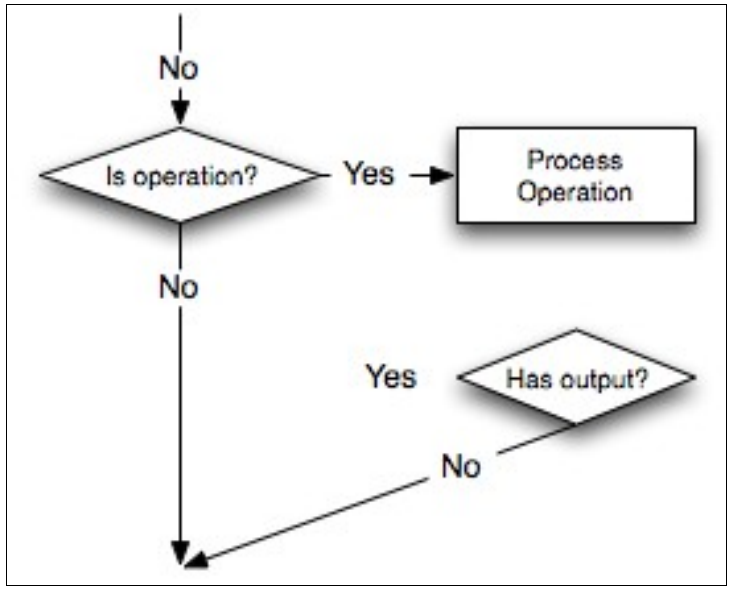
Once in WebGUI, the first thing it has to do is determine if it's dealing with files like extras, uploads, and passthrus. If that's the case, it processes those and then hands it back to mod_perl to serve the files.



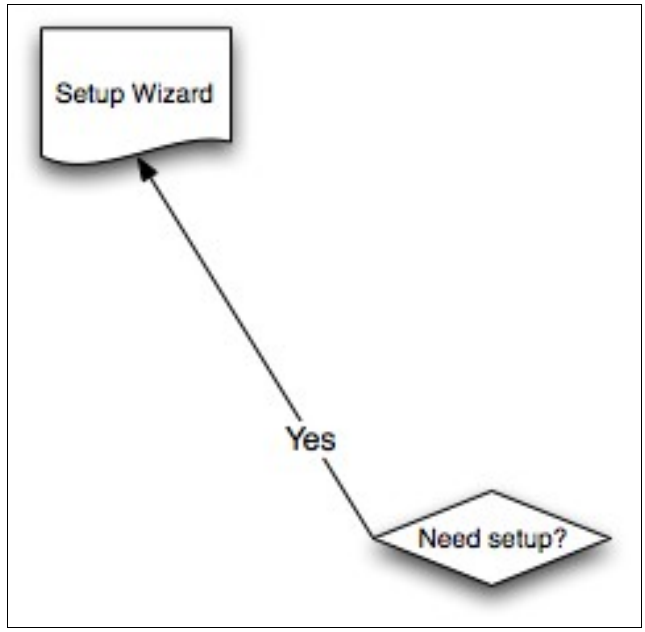
After it's dealt with that, it opens up a session so it can get database access, then determine if the site is being upgraded currently. If it is, it will want to put up a maintenance page.



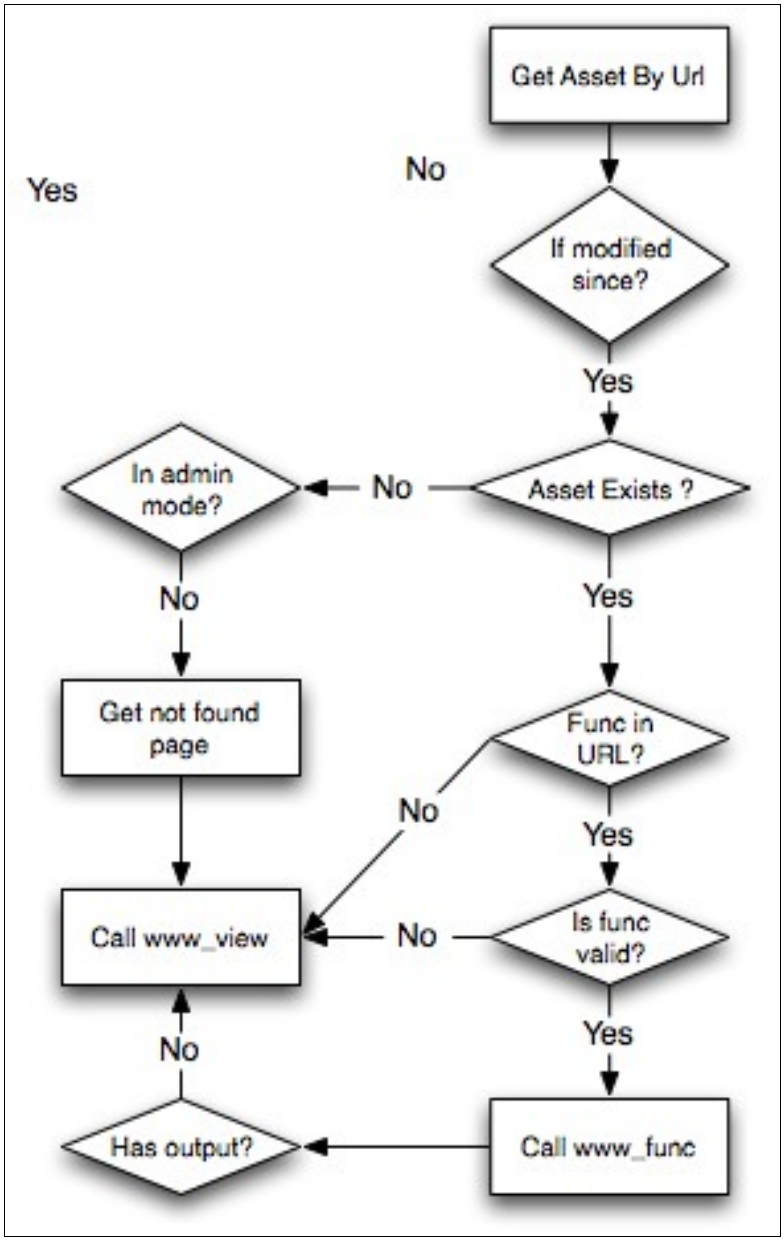
If the site is not being upgraded, it can then process operations. Operations are things like login, turn admin mode on, user management, etc.



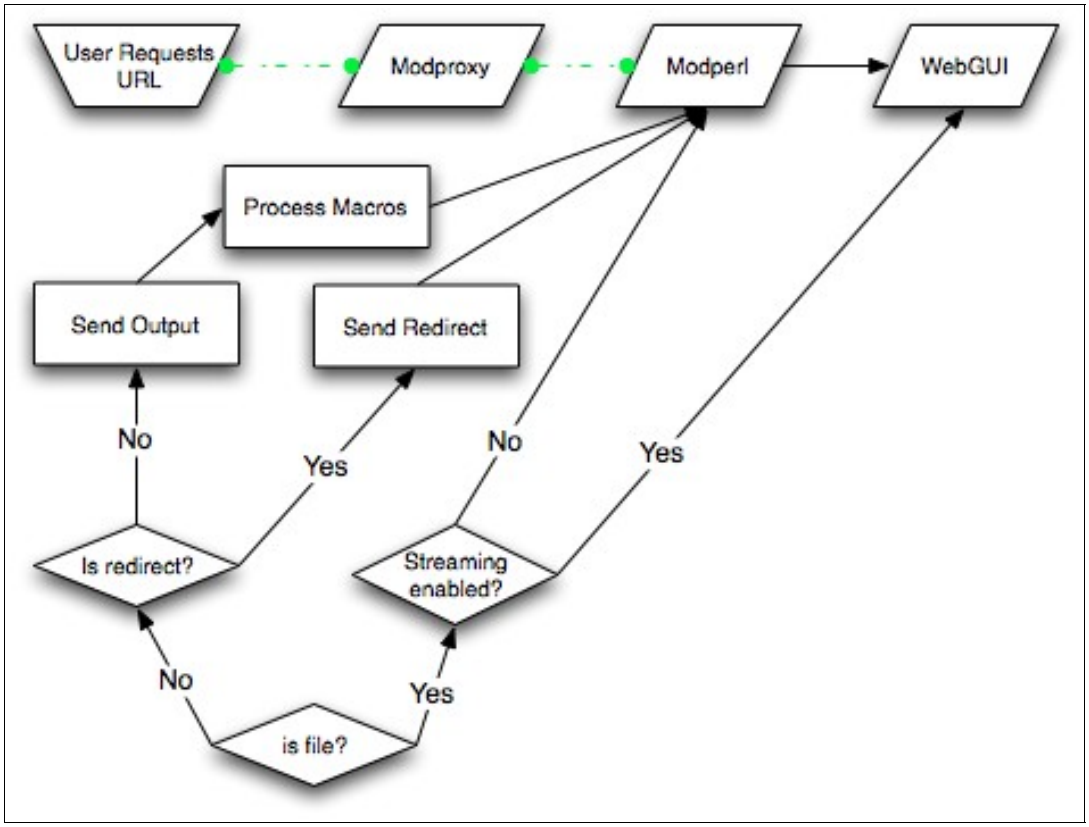
Then, it checks to see if the site needs to be run through the initial configuration wizard (aka setup). If it does, it'll return the setup system.



Once it's gotten through all that, it can finally start looking at serving up content. It gets the asset by its URL, and then process functions on it, if any. It also now has to check to see that the URL exists, and if not it displays a not found message, or an add page message.



And finally, it can do some post processing on the content. It determines if the asset or operation returned a file, a redirect, or basic content. If it was basic content it processes macros on it.

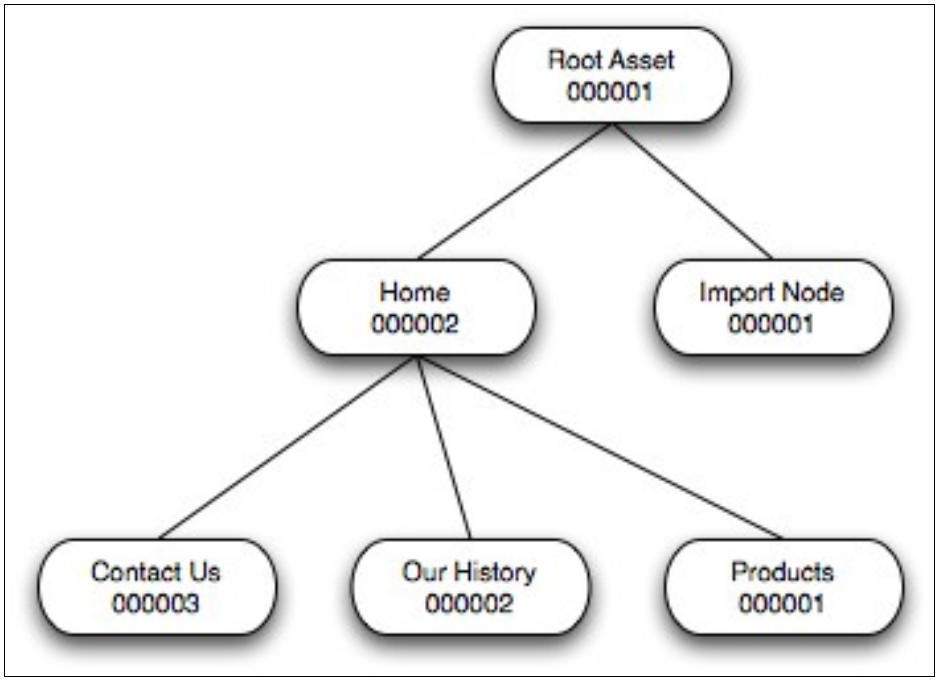


After that, mod_perl hands the data back to mod_proxy, which then sends the request back to the requester.

Lineage

Lineage is the way assets relate themselves to each other. Lineage determines where assets reside in the asset tree, and how they are positioned relative to their siblings.

Each asset has a rank, which is a 6 digit number, left padded by zeros. The lineage of each asset is its rank appended to its parent's lineage.



The lineage of the “Our History” page in this example is “000001000002000002”. Knowing this little piece of information you can infer all kinds of other information. You know that this asset is the third generation: three sets of six digits. Also, the parent's lineage is “000001000002”: just subtract the rank from the lineage. The grandparent's lineage is “000001”: subtract six digits from the parent's lineage.

If you want to see this, run the following command from the MySQL command line:

```
select assetId, className, lineage from asset order by lineage
```

Why is this important? For a number of reasons:

- You can use it to troubleshoot performance problems, especially those related to navigation.
- You can use it to write SQL that you might need for reports or troubleshooting.
- You can use it to fix problems on your site.

Navigation Performance

Navigation performance certainly has to do with how your navigation is configured, but it can also have a lot to do with how your asset tree is built. For example, building a tree right off the root of the system isn't usually a good idea. The Import Node is also right off the root, and depending upon how your navigation is configured, that can mean serious performance problems because your navigation may also be reading the import node's data, which it of course will never need.

When you select “relative to current page” or “relative to root” style navigations, the navigation system will use lineage to determine the start point, and therefore position in the tree matters. If you want to circumvent this all together, you can specify the URL of your start point, and that will give you the highest performing navigation. However, if you want to use a dynamically determined start point, then just be certain that your lineage will support you in that decision. Here's how:

- Put strict limits on your navigation that will not force it to traverse the lineage to places that contain lots of non-displayable assets (import node, collaboration systems, wiki's, etc).
- Use folders to help structure your lineage so that your starting lineage doesn't have to look at a lot of siblings.

SQL and Lineage

One of the great features of the lineage system is that you can get back an entire tree of data with a simple SQL query, something that's not possible with a traditional recursive parent-child relationship. In the previously provided example you can request the entire asset tree like this:

```
select assetId, className, lineage from asset order by lineage
```

You can also get back a portion of a tree. For example, say you have an assetId ('XXX'), and you want to get all of the descendants of that child. You first need to find out the lineage of that asset:

```
select lineage from asset where assetId='XXX'
```

Then you can ask the database to give you all the assetIds that have that lineage ('00001000046'):

```
select assetId from asset where lineage like '00001000046%'
```

You can even combine those two queries using a sub select:

```
select assetId from asset where lineage like ( select concat(lineage,'%') from asset where assetId='XXX' )
```

Fixing Lineage

In very rare circumstances lineage can become corrupted. This happens either if a bug was introduced into the system, or if someone was manually tampering with the database. Luckily, this is easily corrected using the rebuildLineage.pl utility. Type the following command:

```
cd /data/WebGUI/sbin
perl rebuildLineage.pl --help
```

```
Usage: perl rebuildLineage.pl --configfile=<webguiConfig>

This utility will rebuild your WebGUI Lineage Tree. The lineage tree is an
index that is used to make WebGUI run faster. It will also detect and fix
orphan data, and detect cirular relationships in your tree.

WARNING: Use this tool only if you know what you're doing. It should only
be used if somehow your lineage tree has become corrupt (very rare) or if
you have done some massive reorganization of your asset tree and you want
to fill in the gaps between the ranks of your assets. A side effect of
using this utility can be that your assets may no longer be in the same rank
as they once were, which means that they may appear out of order in your
navigation.

    --configFile    WebGUI config file.

Options:

    --help          Display this help message and exit.
```

--quiet Disable output unless there's an error.

To fix the lineage, run this command:

```
cd /data/WebGUI/sbin
perl rebuildLineage.pl --config=www.example.com.conf
```

It will output something like this:

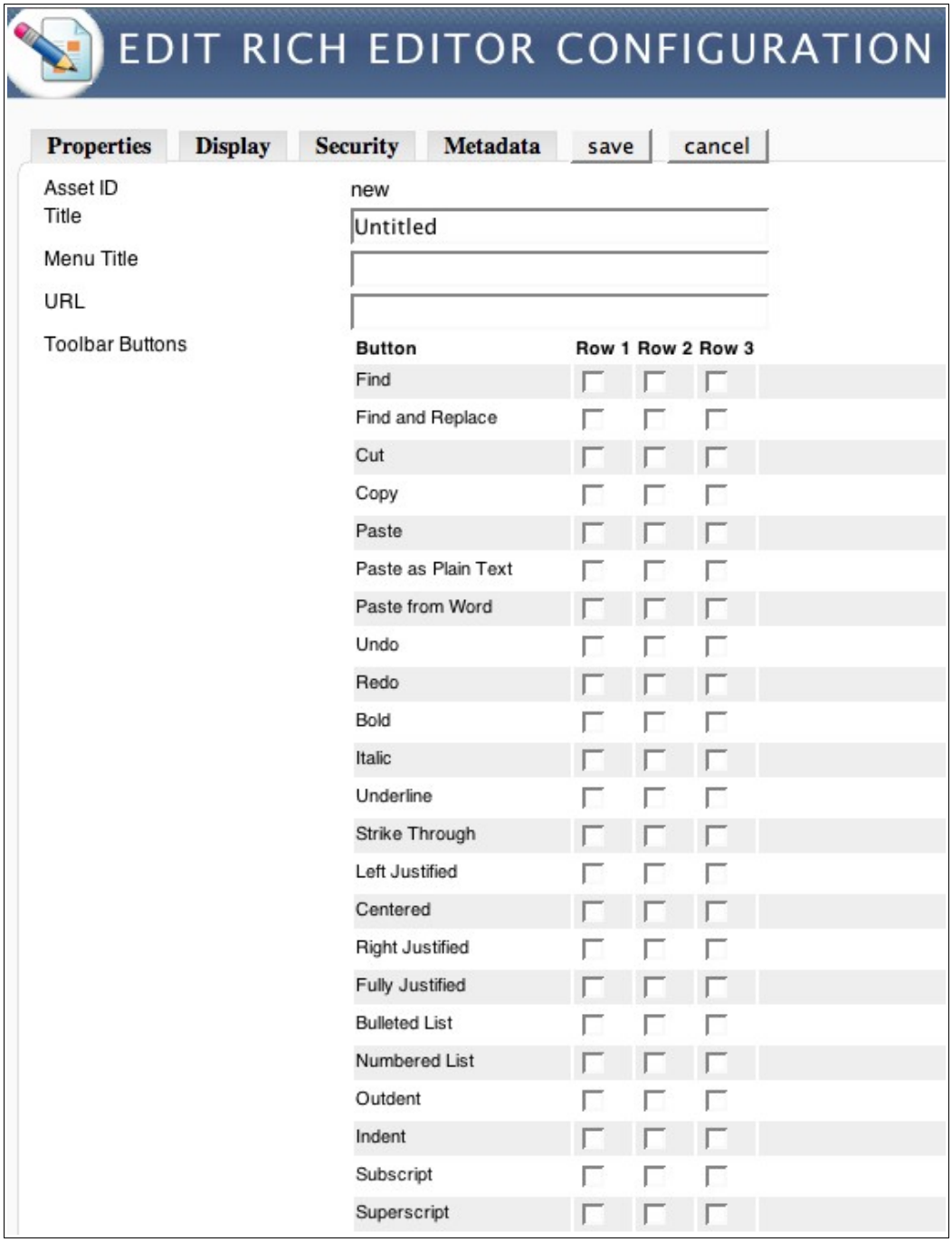
```
Starting...OK
Looking for descendant replationships...
Got the relationships.

Looking for orphans...
No orphans found.

Rewriting existing lineage...
Rebuilding lineage...
Asset ID                 Old Lineage                                    New Lineage
PBasset000000000000001   old___000001                                   000001
PBasset000000000000002   old___000001000001                             000001000001
pbproto000000000000002   old___000001000001000015                       000001000001000001
nbSrhXZQuxIjhWFaFPSuVA   old___000001000001000018                       000001000001000002
PBtmpl0000000000000001   old___000001000001000018000001                 000001000001000002000001
PBtmplHelp000000000001   old___000001000001000018000002                 000001000001000002000002
TvOZs8U1kRXLtwtmyW75pg   old___000001000001000019                       000001000001000003
PBtmpl0000000000000103   old___000001000001000019000001                 000001000001000003000001
PBtmpl0000000000000084   old___000001000001000019000002                 000001000001000003000002
PBtmpl0000000000000002   old___000001000001000019000003                 000001000001000003000003
PBtmpl0000000000000115   old___000001000001000019000004                 000001000001000003000004
PBtmpl0000000000000123   old___000001000001000019000005                 000001000001000003000005
PBtmpl0000000000000129   old___000001000001000019000006                 000001000001000003000006
PBtmpl0000000000000207   old___000001000001000019000007                 000001000001000003000007
xSmREZO3GNzK3M5PaueOOQ   old___000001000001000020                       000001000001000004
PBtmpl0000000000000004   old___000001000001000020000001                 000001000001000004000001
0bx-xoL8TSXXubFuqKAoVQ   old___000001000001000021                       000001000001000005
PBtmpl0000000000000005   old___000001000001000021000001                 000001000001000005000001
taX2UYkFF21ALpFZY2rhMw   old___000001000001000022                       000001000001000006
PBtmpl0000000000000006   old___000001000001000022000001                 000001000001000006000001
K0q_N885Httqev1VCqUWxg   old___000001000001000023                       000001000001000007
PBtmpl0000000000000010   old___000001000001000023000001                 000001000001000007000001
fq1ZkYhH24R5tb96kuT10Q   old___000001000001000024                       000001000001000008
PBtmpl0000000000000011   old___000001000001000024000001                 000001000001000008000001
oHk7fAFhEEkB7dHzi0QOQA   old___000001000001000025                       000001000001000009
PBtmpl0000000000000012   old___000001000001000025000001                 000001000001000009000001
9M-lrlPQWeeNWfvnDnK_Xg   old___000001000001000026                       000001000001000010
PBtmpl0000000000000013   old___000001000001000026000001                 000001000001000010000001
_gBYAdTcbkiyamnqi2Xskg   old___000001000001000027                       000001000001000011
PBtmpl0000000000000014   old___000001000001000027000001                 000001000001000011000001
GNOAsX98vCsl0JRwfwL-gg   old___000001000001000028                       000001000001000012
PBtmpl0000000000000066   old___000001000001000028000001                 000001000001000012000001
PBtmpl0000000000000080   old___000001000001000028000002                 000001000001000012000002
...
68sKwDgf9cGH58-NZcU4lg   old___000001000002                             000001000002
_iHetEvMQUOoxS-T2CM0sQ   old___000001000002000003                       000001000002000001
bX5rYxb6tZ9docY6sUhBlw   old___000001000002000003000001                 000001000002000001000001
Vzv1pWpg_w6R_o-b0rM2qQ   old___000001000002000003000002                 000001000002000001000002
NK8bqlwVRILJknqeCDPBHg   old___000001000002000003000003                 000001000002000001000003
8Bb8gu-me2mhL3ljFyiWLg   old___000001000002000004                       000001000002000002
6QuS-0rosuZTdTv11fobig   old___000001000002000004000001                 000001000002000002000001
FOvmwGC0GtZo5VTxJIL3OA   old___000001000002000004000002                 000001000002000002000002
ix1p0AbwKAz8QWB-T-HHfg   old___000001000002000004000003                 000001000002000002000003
iCYOjohB9SKvAPr6bXElKA   old___000001000002000004000004                 000001000002000002000004
4Yfz9hqBqM8OYMGuQK8oLw   old___000001000002000004000005                 000001000002000002000005
Wl8WZ43g2rK5AYr9o4zY7w   old___000001000002000004000006                 000001000002000002000006
LBuiKzg2mWwmOPS9AgV3bg   old___000001000002000004000007                 000001000002000002000007
jTNggl7AoVSUc_ZzrvuCmw   old___000001000002000004000008                 000001000002000002000008
2TqQc4OISddWCZmRY1_m8A   old___000001000002000005                       000001000002000003
```

```
fK-HMSboA3uu0c1KYkYspA   old___000001000002000005000001                    000001000002000003000001
Swf6L8poXKc7hUaNPkBevw   old___000001000002000006                          000001000002000004
Szs5eev3OMssmnsyLRZmWA   old___000001000002000006000001                    000001000002000004000001
x3OFY6OJh_qsXkZfPwug4A   old___000001000002000007                          000001000002000005
pJd5TLAjfWMVXD6sCRLwUg   old___000001000002000007000001                    000001000002000005000001
OhdaFLE7sXOzo_SIP2ZUgA   old___000001000002000008                          000001000002000006
m4YJFaqzultnB_sj1Uq0aw   old___000001000002000009                          000001000002000007
IWFxZDyGhQ3-SLZhELa3qw   old___000001000002000010                          000001000002000008
JACIChYiCgzZzcYeyJ-s0w   old___000001000002000011                          000001000002000009
PBasset000000000000003   old___000001000003                                000001000003
tempspace0000000000000   old___000001000004                                000001000004
Cleaning up...OK

Don't forget to clear your cache.
```

When it's finished running, clear your cache. Your cache will likely no longer match what's in the database, so not clearing your cache could cause all kinds of corruption.

Rich Editor

WebGUI's rich editor is highly configurable so that you can make it work exactly as you need it to for your site.



The rich editor can do almost everything you can do in a word processing program, including text formatting, image insertion, tables, lists, and more. In addition, it's integrated into WebGUI so that you can use WebGUI's resources from within the rich editor, like linking to other files and pages, and inserting WebGUI images.

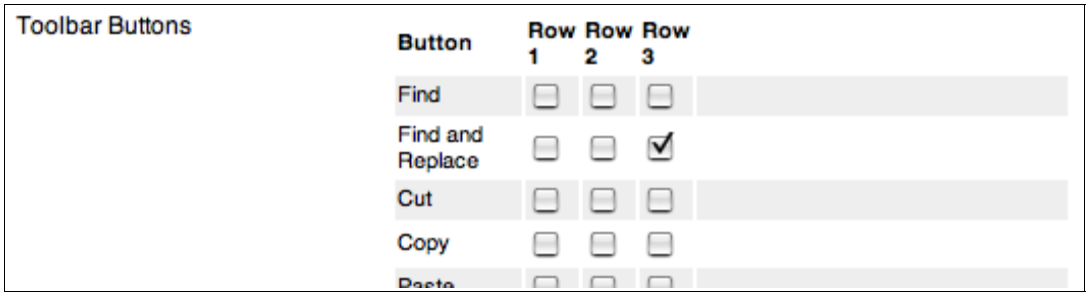
Configuration

1. To configure the rich editor go to Admin Console > Assets.

2. And then select Search.

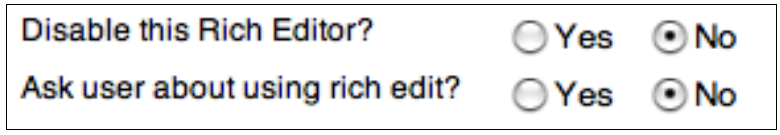
3. Now search for the “WebGUI::Asset::RichEdit” class without any keywords.

4. That search will give you a list of all the rich editors that are defined in WebGUI. Click “Edit” on one of the rich editors to edit its properties. Among other things, the edit screen will display a list of buttons you can add to the toolbar in the rich editor.

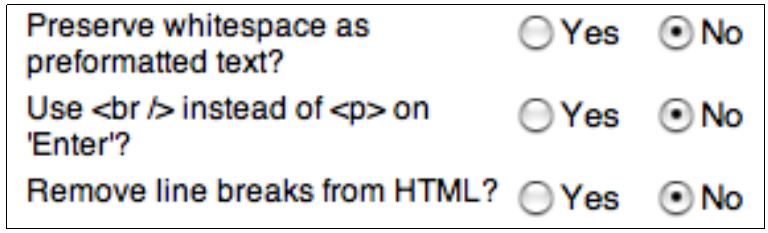


5. You have three columns of checkboxes, which will allow you to choose which toolbar buttons are displayed, and on which row of the toolbar.

6. Some power users will prefer to either not use the rich editor, or to decide at the time of editing whether to edit the HTML directly, or to enable the rich editor. These options will give power users those abilities.



7. The rich editor also gives properties for formatting. When “Preserve whitespace as preformatted text?” is used, spaces pasted or typed into the editor will be preserved as formatted content. When the “
 instead of <p>” option is enabled the editor will use <br /> in the resulting HTML instead of <p> tags, and you'll have to type the carriage return twice to make room for the next paragraph. The “Remove line breaks from HTML?” option is similar to the whitespace option above, except that instead of stripping out extra spaces, it strips out extra carriage returns from the HTML.



8. The “Text Direction” option allows you to specify which way the text flows in the rich editor, which is useful for many non-western languages.

9. The “CSS File” option allows you to associate a cascading style sheet file with the rich editor. When you do this in combination with the “Apply Style” toolbar option, your users can apply formatting to their content using styles you create for them with CSS.

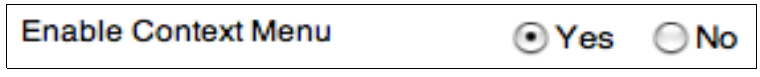
You need to use a full URL like one of the following:

```
/styles/site.css

http://www.example.com/site.css

^FileUrl(some/file/asset);
```

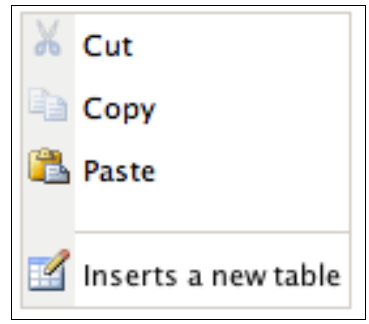
Note that the URL can be the fully qualified url to a snippet acting like a CSS file. However, it cannot be a relative url like this:

```
site.css

style/site.css
```

10. The “Enable Context Menu” option creates a context menu in the rich editor when the user right clicks.

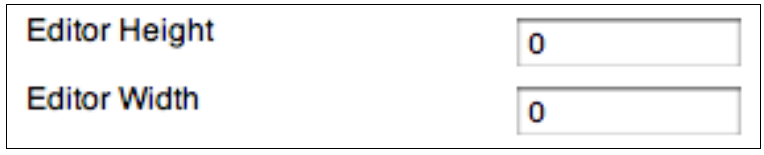


11. The context menu looks like this:

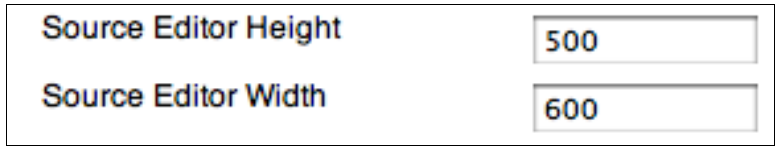
    

    Since it is a context menu, however, it will have different options depending upon what the user right clicks on. This won't work on all browsers, because some browsers don't allow javascript triggered right clicks.

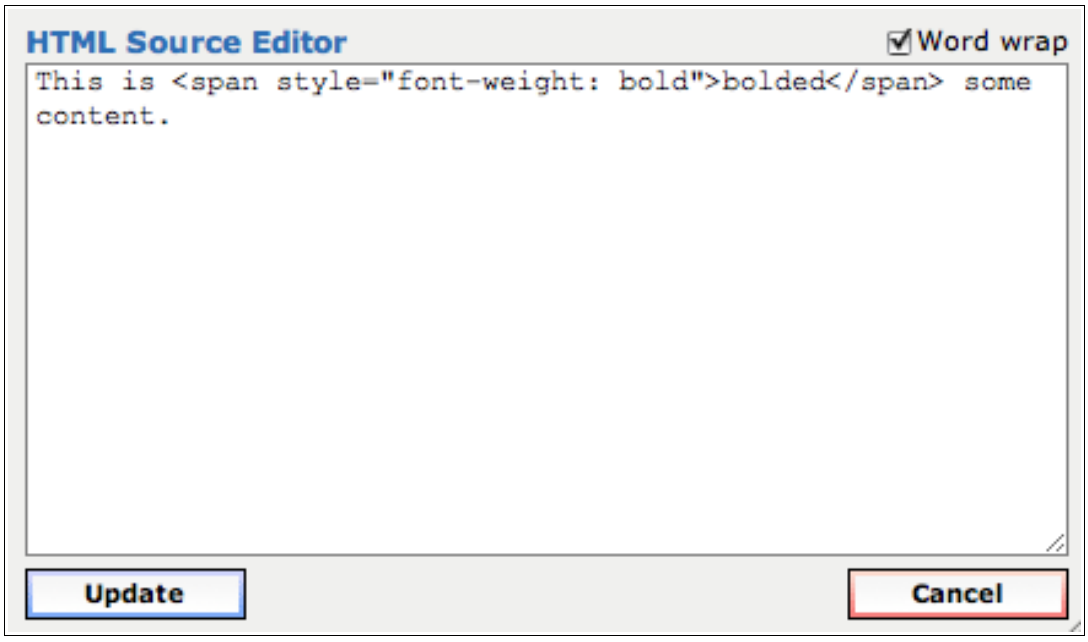
12. Under the Display tab you can set how big this editor will appear by default on whatever page it is rendered. Setting it to 0 will cause it to try to stretch to fit the area it's given.

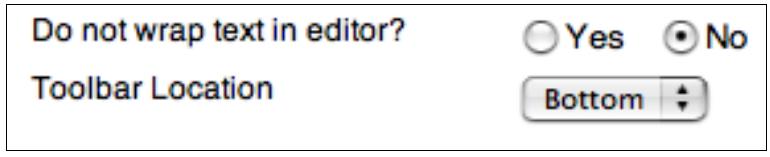
13. You can also set the width and height of the HTML source editor. That editor is enabled as a toolbar button.

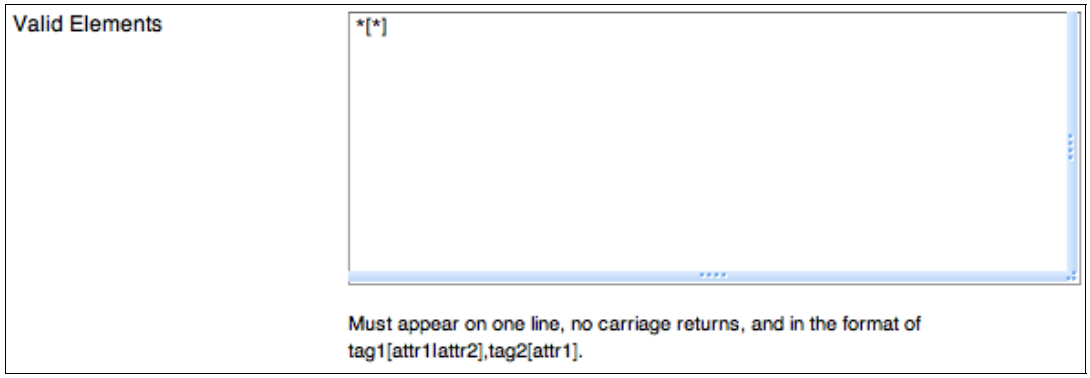
    The HTML Source Editor looks like this.



14. You can also tell the editor whether or not to wrap text as it is typed, or to create scroll bars instead. You can also determine whether the toolbar should be at the top or bottom of the editor.



15. On the Security tab you have one final option.



You can use this option to enforce what kind of HTML is allowed to

be created by your users. The default is:

```
*[*]
```

This means that the users can create any tags they want, and those tags can have any attributes that they want to put in them. If you wanted to allow users to create any tags they want, but only allow title and class attributes, then you could configure Valid Elements as follows:

*[title|class]

If you want to use a typical HTML rule set you could do something like:

```
+a[id|style|rel|rev|charset|hreflang|dir|lang|tabindex|accesskey|type|name|href|target|title|
class|onfocus|onblur|onclick|ondblclick|onmousedown|onmouseup|onmouseover|
onmousemove|onmouseout|onkeypress|onkeydown|onkeyup],-strong/-b[class|style],-em/-
i[class|style],-strike[class|style],-u[class|style],#p[id|style|dir|class|align],-ol[class|style],-
ul[class|style],-li[class|style],br,img[id|dir|lang|longdesc|usemap|style|class|src|
onmouseover|onmouseout|border|alt=|title|hspace|vspace|width|height|align],-sub[style|
class],-sup[style|class],-blockquote[dir|style],-table[border=0|cellspacing|cellpadding|width|
height|class|align|summary|style|dir|id|lang|bgcolor|background|bordercolor],-tr[id|lang|dir|
class|rowspan|width|height|align|valign|style|bgcolor|background|bordercolor],tbody[id|
class],thead[id|class],tfoot[id|class],-td[id|lang|dir|class|colspan|rowspan|width|height|align|
valign|style|bgcolor|background|bordercolor|scope],-th[id|lang|dir|class|colspan|rowspan|
width|height|align|valign|style|scope],caption[id|lang|dir|class|style],-div[id|dir|class|align|
style],-span[style|class|align],-pre[class|align|style],address[class|align|style],-h1[id|style|dir|
class|align],-h2[id|style|dir|class|align],-h3[id|style|dir|class|align],-h4[id|style|dir|class|align],-
h5[id|style|dir|class|align],-h6[id|style|dir|class|align],hr[class|style],-font[face|size|style|id|
class|dir|color],dd[id|class|title|style|dir|lang],dl[id|class|title|style|dir|lang],dt[id|class|title|
style|dir|lang]
```

And if you want to do XHTML Strict, you could do:

```
a[accesskey|charset|class|coords|dir<ltr?rtl|href|hreflang|id|lang|name|onblur|onclick|
ondblclick|onfocus|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|rel|rev|shape<circle?default?poly?rect|style|
tabindex|title|target|type],abbr[class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],acronym[class|dir<ltr?rtl|id|id|lang|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],address[class|align|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],applet[align<bottom?left?middle?right?top|alt|archive|class|code|
codebase|height|hspace|id|name|object|style|title|vspace|width],area[accesskey|alt|class|
coords|dir<ltr?rtl|href|id|lang|nohref<nohref|onblur|onclick|ondblclick|onfocus|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|shape<circle?default?poly?rect|style|tabindex|title|target],base[href|
target],basefont[color|face|id|size],bdo[class|dir<ltr?rtl|id|lang|style|title],big[class|dir<ltr?rtl|
id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],blockquote[dir|style|cite|class|dir<ltr?rtl|
```

id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],body[alink|background|bgcolor|class|
dir<ltr?rtl|id|lang|link|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onload|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|onunload|style|title|
text|vlink],br[class|clear<all?left?none?right|id|style|title],button[accesskey|class|dir<ltr?rtl|
disabled<disabled|id|lang|name|onblur|onclick|ondblclick|onfocus|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|
tabindex|title|type|value],caption[align<bottom?left?right?top|class|dir<ltr?rtl|id|lang|
onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],center[class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],cite[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],code[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],col[align<center?char?justify?left?right|char|charoff|
class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|span|style|title|
valign<baseline?bottom?middle?top|width],colgroup[align<center?char?justify?left?right|
char|charoff|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|span|style|title|
valign<baseline?bottom?middle?top|width],dd[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],del[cite|class|datetime|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],dfn[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],dir[class|compact<compact|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],div[align<center?justify?left?right|class|dir<ltr?rtl|id|
lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],dl[class|compact<compact|dir<ltr?rtl|id|
lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],dt[class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],em/i[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],fieldset[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],font[class|color|dir<ltr?rtl|face|id|lang|size|style|
title],form[accept|accept-charset|action|class|dir<ltr?rtl|enctype|id|lang|method<get?post|
name|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|onreset|onsubmit|style|title|target],frame[class|
frameborder|id|longdesc|marginheight|marginwidth|name|noresize<noresize|
scrolling<auto?no?yes|src|style|title],frameset[class|cols|id|onload|onunload|rows|style|
title],h1[align<center?justify?left?right|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],h2[align<center?justify?left?right|class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],h3[align<center?justify?left?right|class|dir<ltr?rtl|id|
lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],h4[align<center?justify?left?right|class|

dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|
title],h5[align<center?justify?left?right|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],h6[align<center?justify?left?right|class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],head[dir<ltr?rtl|lang|profile],hr[align<center?left?right|
class|dir<ltr?rtl|id|lang|noshade<noshade|onclick|ondblclick|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|size|style|
title|width],html[dir<ltr?rtl|lang|version],iframe[align<bottom?left?middle?right?top|class|
frameborder|height|id|longdesc|marginheight|marginwidth|name|scrolling<auto?no?yes|src|
style|title|width],img[align<bottom?left?middle?right?top|alt|border|class|dir<ltr?rtl|height|
hspace|id|ismap<ismap|lang|longdesc|name|onclick|ondblclick|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|src|style|
title|usemap|vspace|width],input[accept|accesskey|align<bottom?left?middle?right?top|alt|
checked<checked|class|dir<ltr?rtl|disabled<disabled|id|ismap<ismap|lang|maxlength|
name|onblur|onclick|ondblclick|onfocus|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|onselect|readonly<readonly|size|src|
style|tabindex|title|
type<button?checkbox?file?hidden?image?password?radio?reset?submit?text|usemap|
value],ins[cite|class|datetime|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|
title],isindex[class|dir<ltr?rtl|id|lang|prompt|style|title],kbd[class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],label[accesskey|class|dir<ltr?rtl|for|id|lang|onblur|
onclick|ondblclick|onfocus|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],legend[align<bottom?left?right?top|
accesskey|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|title],li[class|
dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|title|type|value],link[charset|
class|dir<ltr?rtl|href|hreflang|id|lang|media|onclick|ondblclick|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|rel|rev|
style|title|target|type],map[class|dir<ltr?rtl|id|lang|name|onclick|ondblclick|onkeydown|
onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|
onmouseup|style|title],menu[class|compact<compact|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],meta[content|dir<ltr?rtl|http-equiv|lang|name|
scheme],noframes[class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|
title],noscript[class|dir<ltr?rtl|id|lang|style|title],object[align<bottom?left?middle?right?top|
archive|border|class|classid|codebase|codetype|data|declare|dir<ltr?rtl|height|hspace|id|
lang|name|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|standby|style|tabindex|title|type|
usemap|vspace|width],ol[class|compact<compact|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|start|style|title|type],optgroup[class|dir<ltr?rtl|disabled<disabled|
id|label|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|title],option[class|dir<ltr?rtl|
disabled<disabled|id|label|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|selected<selected|
style|title|value],p[align<center?justify?left?right|class|dir<ltr?rtl|id|lang|onclick|ondblclick|

onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],param[id|name|type|value|
valuetype<DATA?OBJECT?REF],pre/listing/plaintext/xmp[align|class|dir<ltr?rtl|id|lang|
onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title|width],q[cite|class|dir<ltr?rtl|id|lang|
onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],s[class|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],samp[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],script[charset|defer|language|src|type],select[class|
dir<ltr?rtl|disabled<disabled|id|lang|multiple<multiple|name|onblur|onchange|onclick|
ondblclick|onfocus|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|size|style|tabindex|title],small[class|dir<ltr?rtl|id|
lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],span[align<center?justify?left?right|class|
dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|title],strike[class|class|
dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|title],strong/b[class|dir<ltr?rtl|
id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title],style[dir<ltr?rtl|lang|media|title|
type],sub[class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|title],sup[class|
dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|style|
title],table[align<center?left?right|bgcolor|border|cellpadding|cellspacing|class|dir<ltr?rtl|
frame|height|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|
onmousemove|onmouseout|onmouseover|onmouseup|rules|style|summary|title|
width],tbody[align<center?char?justify?left?right|char|class|charoff|dir<ltr?rtl|id|lang|
onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|
onmouseout|onmouseover|onmouseup|style|title|
valign<baseline?bottom?middle?top],td[abbr|align<center?char?justify?left?right|axis|
bgcolor|char|charoff|class|colspan|dir<ltr?rtl|headers|height|id|lang|nowrap<nowrap|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|rowspan|scope<col?colgroup?row?rowgroup|style|title|
valign<baseline?bottom?middle?top|width],textarea[accesskey|class|cols|dir<ltr?rtl|
disabled<disabled|id|lang|name|onblur|onclick|ondblclick|onfocus|onkeydown|onkeypress|
onkeyup|onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|onselect|
readonly<readonly|rows|style|tabindex|title],tfoot[align<center?char?justify?left?right|char|
charoff|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|title|
valign<baseline?bottom?middle?top],th[abbr|align<center?char?justify?left?right|axis|
bgcolor|char|charoff|class|colspan|dir<ltr?rtl|headers|height|id|lang|nowrap<nowrap|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|rowspan|scope<col?colgroup?row?rowgroup|style|title|
valign<baseline?bottom?middle?top|width],thead[align<center?char?justify?left?right|char|
charoff|class|dir<ltr?rtl|id|lang|onclick|ondblclick|onkeydown|onkeypress|onkeyup|
onmousedown|onmousemove|onmouseout|onmouseover|onmouseup|style|title|
valign<baseline?bottom?middle?top],title[dir<ltr?rtl|lang],tr[abbr|
align<center?char?justify?left?right|bgcolor|char|charoff|class|rowspan|dir<ltr?rtl|id|lang|
onclick|ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|

```
onmouseout|onmouseover|onmouseup|style|title|
valign<baseline?bottom?middle?top],tt[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],u[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title],ul[class|compact<compact|dir<ltr?rtl|id|lang|onclick|
ondblclick|onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title|type],var[class|dir<ltr?rtl|id|lang|onclick|ondblclick|
onkeydown|onkeypress|onkeyup|onmousedown|onmousemove|onmouseout|
onmouseover|onmouseup|style|title]
```

Of course that's a terribly ridiculous, but useful, example. The intent was to show you exactly how strict you can make the HTML produced by the rich editor.

You can create whatever rules you want. You need to add all your rules on a single line (no carriage returns). The basic format is

```
tagname[attribute1|attribute2],tag2[attrib1,attrib2]
```

Here are all the control characters for this mini language:

```
,
```

Separates element chunk definitions.

```
/
```

Separates element synonymous. The first element is the one that will be output.

```
|
```

Separates attribute definitions.

```
[
```

Starts a new attribute list for an element definition.

```
]
```

Ends an attribute list for an element definition.

```
=
```

Makes the attribute default to the specified value. For example, "target=_blank"

```
:
```

Forces the attribute to the specified value. For example, "border:0"

<

Verifies the value of an attribute. For example, "target<_blank?_self"

?

Separates attribute verification values. See above.

+

Makes the element open if no child nodes exists. For example, "+a".

-

Enables removal of empty elements such as . For example, "-strong".

#

Enables padding of empty elements. This will pad with   if they are empty. For example, "#p".

!

Makes attributes required. If none of the required attributes are set, the element will be removed. For example, "!href".

{$uid}

Results in a unique ID. For example, "p[id:{$uid}]".

In addition, wildcards such as *,+,? may be used in element or attribute name matching.

Workflows

The WebGUI workflow system controls the flow of content through the system. Because the workflow is customizable, you can control what actions are performed, and when those actions take place. Sometimes viewing an existing workflow on the site helps envision how all the activities come together.

To view current workflows in the system, select Workflow from the Admin Console. This will open the “Manage Workflows” screen.

This screen contains the name of each existing workflow, an indication of if the workflow is currently enabled, and icons to delete or edit each workflow. To edit an existing workflow, simply simply click on its edit button. This will open the “Edit Workflow” screen.

At the top of the screen is the “Workflow ID.” This is a unique identifier generated by WebGUI at the time of workflow creation. Next is the “Object Type.” In this case, it is WebGUI::Version Tag, which means this workflow can be applied to version tags. The “Title” is simply the name of the workflow, and the “Description” field contains an explanation of the workflow’s function. The “Is Enabled?” toggle indicates if the workflow is enabled for use.

The “Mode” field currently has four options: Serial, Parallel, Realtime, and Singleton.

- Serial means that if multiple instances of this workflow are created they will be executed one at a time in order. The first in line will be executed, then the next, and so on. This will be executed behind the scenes as the workflow engine has the time and resources to complete the tasks. An example for this workflow mode is checking an email box. If the workflow instances were allowed to execute in a parallel fashion then you might download the same message multiple times. By keeping it serial you will get exactly one copy of each email message.

- Parallel means that multiple instances of this workflow will all be executed simultaneously. This will be executed behind the scenes as the workflow engine has the time and resources to complete the tasks. An example for this workflow mode is content publishing. We want our content published as quickly as possible, we don't care about publishing in the order the content was committed.

- Realtime means that this workflow will be executed while the user waits, instead of letting the workflow engine execute it when it has time and resources. This can be dangerous if the workflow takes a long time to complete because the user may believe that the process has stalled and may refresh the browser, or leave before the process is done. However, from a user's perspective there is also “instant gratification” because whatever work they were expecting to be done, will be done right then and there. An example of this mode is also content publishing, but only if there is no approval process.

- Singleton means that multiple instances of this workflow will not be allowed. If one instance already exists and another is created, the second will be discarded. This will be executed behind the scenes as the workflow engine has the time and resources to complete the tasks. An example of this mode is for sending out email. You know you want to send out all of your email messages. You also know you

are going to try to send out messages every X minutes, so you don't ever need more than one process sending out the messages.

At the bottom of the screen are listed a number of links. These links indicate activities that can be added to the workflow. Simply click on a link, add the appropriate information and save it to the workflow.

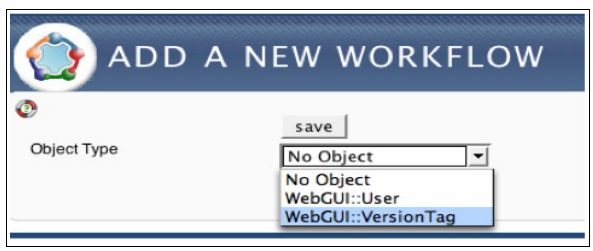
The activities listed in black are already part of the workflow. Next to these are icons that allow you delete an activity, edit it, or change its position in the order of performance.

If someone performs an action on the web site that uses this workflow while it is being edited, the system will use an instance of the workflow that mirrors it as it is being edited. Because of this, the content may not move through the system as expected. Therefore, you should always disable a workflow before editing it, and re-enable the workflow when you're done editing.

Add a Workflow

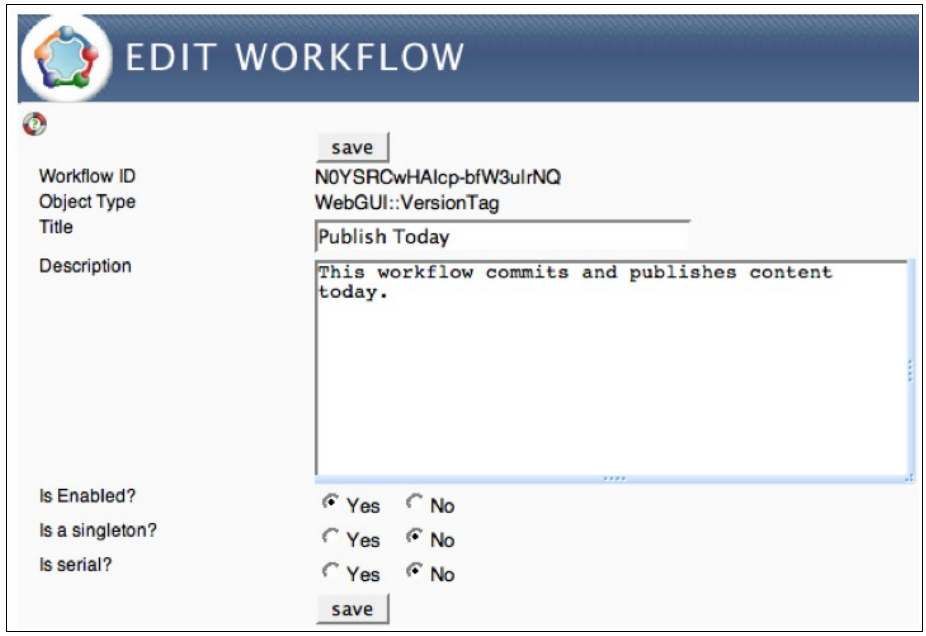
1. To add a new workflow, click on the “Add a new worfklow” link on the far right hand side of the “Manage Workflows” screen. In this example a workflow will be added that commits a version tag on a specific date, in this case the day the workflow is created.

2. In the “Add a New Workflow” screen, select the “Object Type” the workflow will be applied to. “No Object” includes maintenance tasks like decaying karma and cleaning the file cache. “WebGUI::User” applies to a user; “WebGUI::VersionTag” apples to version tags.



3. Click save.

4. The “Edit Workflow” screen will appear.



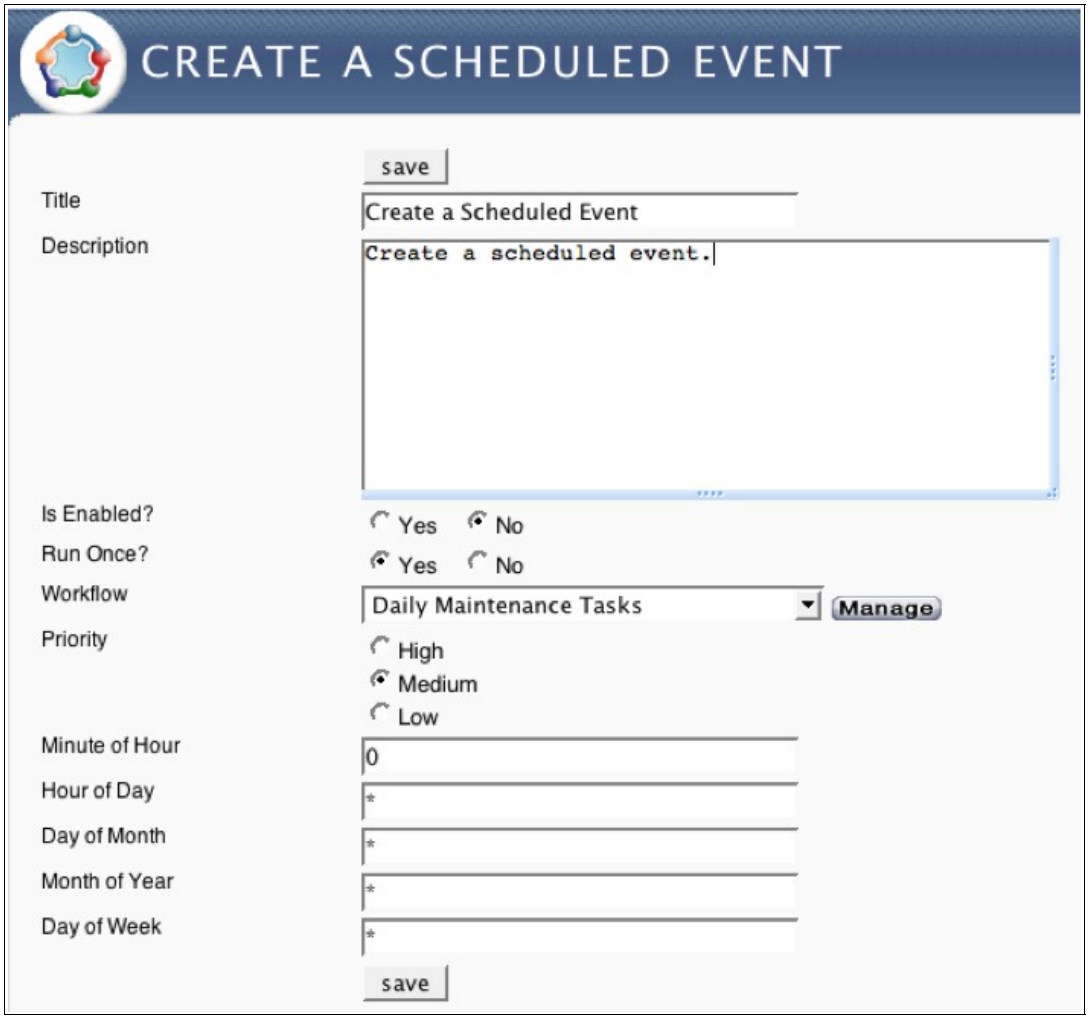
5. Enter a title for the workflow in the “Title” field.

6. Enter a description of the workflow’s function in the “Description” field.

7. Set the “Is Enabled?” toggle to Yes.

8. Select an activity to include in the workflow from the links at the bottom of the screen. In this example, ”Create a Scheduled Event” was selected.

9. A screen will open on which to enter information about the workflow activity you selected. This screen and its fields will differ depending upon which activity you selected.

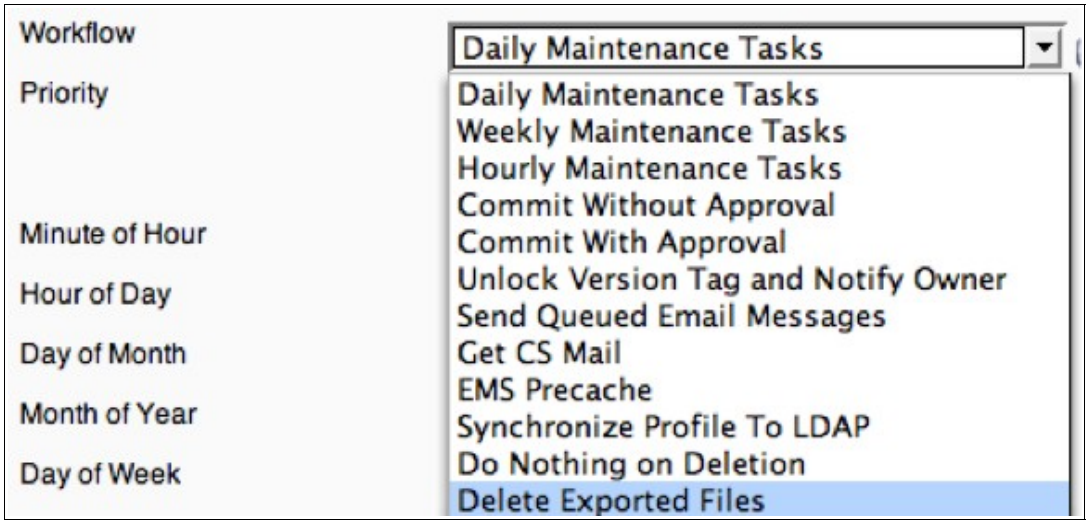


10. You can give the activity a title and enter text in the description field explaining the activity's purpose.

11. Toggle the “Is Enabled?” field to Yes (default setting is No).

12. If the “Run Once?” field is toggled to Yes this task will execute at the scheduled time and then delete itself.

13. Next, select a workflow from the “Workflow” dropdown menu. This indicates the workflow you want to run at the time the event is scheduled. In this example, Commit Without Approval.



14. In the “Priority” field select the option that best indicates the priority of this activity; the higher the priority the faster the activity will be performed.

15. Set the time and date fields to indicate when the event should occur.

16. Click save. You will be returned to the “Edit Workflow” screen where you will notice the task now added to the workflow. This appears at the bottom of the screen in black, with a number of editing icons. You will also see it added on the “Manage Workflows” screen.

Now, you can assign this workflow when manually creating a version tag, or you can edit an existing version tag and assign a workflow to it. To do this, select Version Tags from the Admin Console, and click on the edit button next to the version tag you would like to assign this workflow to in the “Edit Version Tag” screen. This will open a screen from which you may select the appropriate workflow. Also, the workflow can be assigned globally so that any new version tags created will have this workflow applied. To assign the workflow globally, select Settings from the Admin Console and alter the “Default Version Tag Workflow” field. This is where you assign the workflow to be executed on all version tags in the system.

Examples of Workflows

The Object Type you select in the “Add a New Workflow” screen will determine the list of available tasks to add to a workflow in the “Edit Workflow” screen.

Object Type: No Object

Available Workflow Tasks:

- Decay Karma: allows you to set the minimum value a user's karma can decay to, and how many points of karma will be removed when the workflow runs.
- Empty Clipboard to Trash: allows you to establish a length of time content is allowed to sit in the Clipboard before the system removes it to the trash.
- Clean Temp Storage: allows you to set an amount of time to pass before temp files are cleared from the system.
- Clean File Cache: allows you to set how big the file cache can get before WebGUI begins pruning old cache entries.
- Clean Login History: allows you to set an amount of time to pass before WebGUI will begin deleting entries in the Login History.
- Archive Old Threads: when this task is run old threads, based on each Collaboration System's settings, will be archived.
- Trash Expired Events: allows you to set an interval of time to pass before an event in the Calendar is moved to the trash. The default interval is 30 days.
- Create a Scheduled Event: allows you to schedule an event in the system for a specific date and time; an example would be to schedule a version tag to be committed at a specific date and time.
- Delete Expired Sessions: this workflow deletes expired sessions.
- Expire User Groupings: this workflow activity will go through all

groups and handle group expiration notifications and deletions. For example, if a group expiration notification has been set up, this workflow activity will send out the appropriate emails to group members to alert them of the upcoming expiration. Once the group's delete offset has passed, this workflow activity will delete users from the group.

- Purge Old Asset Revisions: this workflow activity allows you to set an amount of time to pass before WebGUI will delete versions of assets stored in the Versioning system. The default setting in WebGUI is to delete versions of assets older than one year.

- Expire Subscription Codes: this activity will go through all subscription codes and expire any subscription code that is unused after the expiration date has passed.

- Purge Old Trash: allows you to set a length of time to pass before assets are deleted from the Trash. WebGUI's default setting is to purge assets from the Trash after 30 days.

- Get Syndicated Content: allows the Syndicated Content asset to serve pages more quickly by pre-fetching syndicated content URL's.

- Process Recurring Payments: this activity will process all recurring payment transactions in the Commerce system. When the activity is finished, an email with the details of those transactions is sent to the user assigned in the Settings to receive such information.

- Sync Profiles to LDAP: this activity will synchronize the profiles of all users configured for LDAP authentication (this runs from LDAP to WebGUI).

- Summarize Passive Profile Log: summarizes passive profiling data for all registered users and then deletes the log.

- Send Queued Mail Messages: processes queued emails. This activity will only run in one minute intervals.

- Clean Database Cache: prunes the size of a database cache based on the user configured cache size and the expiration time of items in the cache. If pruning expired items does not reduce the size of the cache to the value configured by the user, then the expiration time will be increased by 30 minutes and the process will repeat until it meets the size requirement.

- Update Calendar Feeds: fetches iCal feeds for the Calendar asset.

Object Type: WebGUI::User

Available Workflow Tasks:

- Create a Scheduled Event: allows you to schedule an event in the system for a specific date and time; an example would be to schedule a version tag to be committed at a specific date and time.
- Notify About User: sends out an email about a user.

Object Type: WebGUI::Version Tag

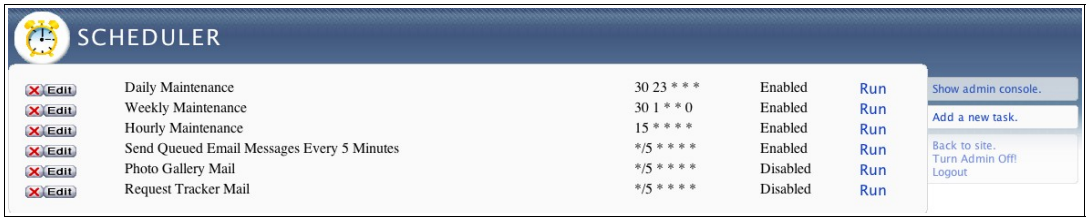
Available Workflow Tasks:

- Commit Version Tag: this activity commits a version tag; in a workflow dealing with committing version tags, this should be the last activity.
- Rollback Version Tag: rolls back a version tag, effectively eliminating all versions of assets contained in that tag.
- Trash Version Tag: moves all assets contained within a version tag to the trash; after this is done, the tag is locked.
- Create a Scheduled Event: allows you to schedule an event in the system for a specific date and time; an example would be to schedule a version tag to be committed at a specific date and time.
- Unlock Version Tag: unlocks a locked version tag so assets can be edited; usually this is used to unlock a committed tag that was rejected in an authorization process.
- Notify About Version Tag: sends out an email containing information about a version tag. The message contains text input when setting up this activity as well as the URL to the first asset in the tag.
- Request Approval for Version Tag: sends out an email to all members of a selected group. The email contains any comments input by commiter as well as the URL to the first asset in the tag. The first user to respond to the email can either approve or deny the commit. If the tag is approved, the tag will be committed, if denied it will move on to the next assigned workflow task.

- Export Version Tag to HTML: exports each asset contained within a version tag to HTML.

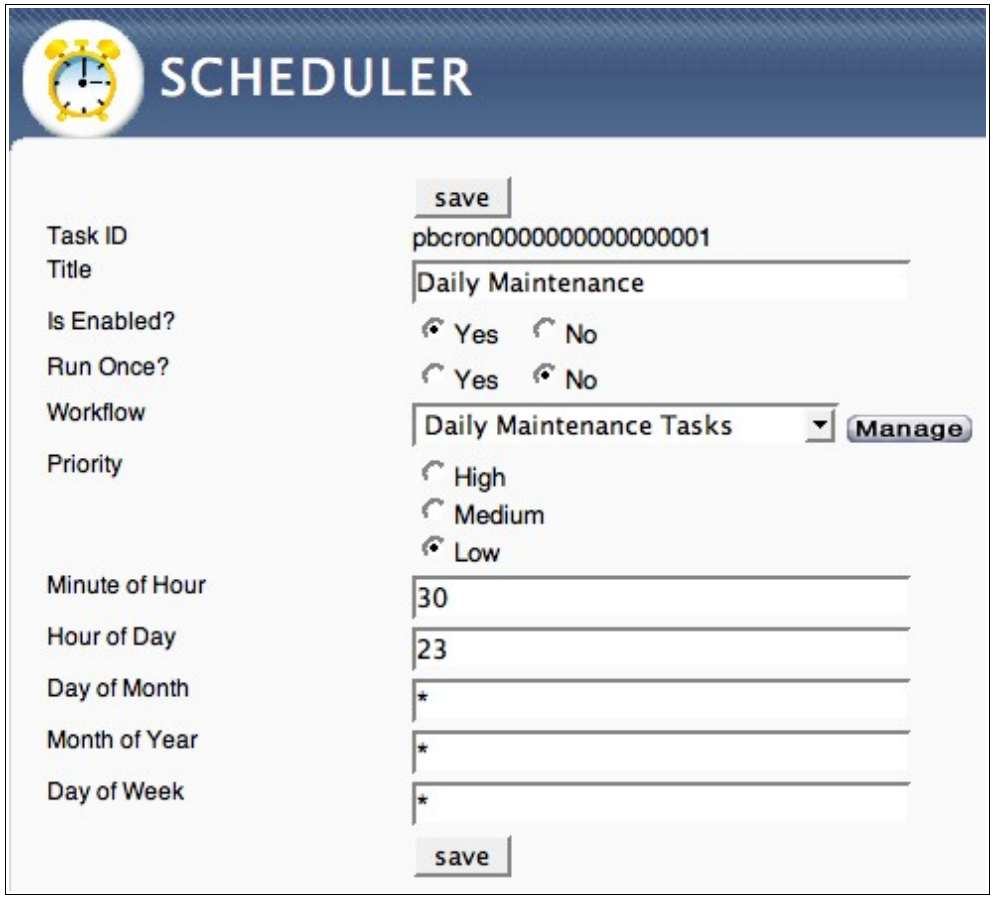
Scheduler

The Scheduler allows you to customize the date and time tasks on the site take place. To access the Scheduler, click on its icon in the Admin Console. This will open the “Scheduler” screen.



On this screen are listed all currently scheduled tasks in the system, the time they are scheduled to run, and whether or not they are currently enabled. To the left of each scheduled task is a toolbar that allows you to either delete or edit a task.

If the edit button of a task is clicked, the main screen for that task will open.



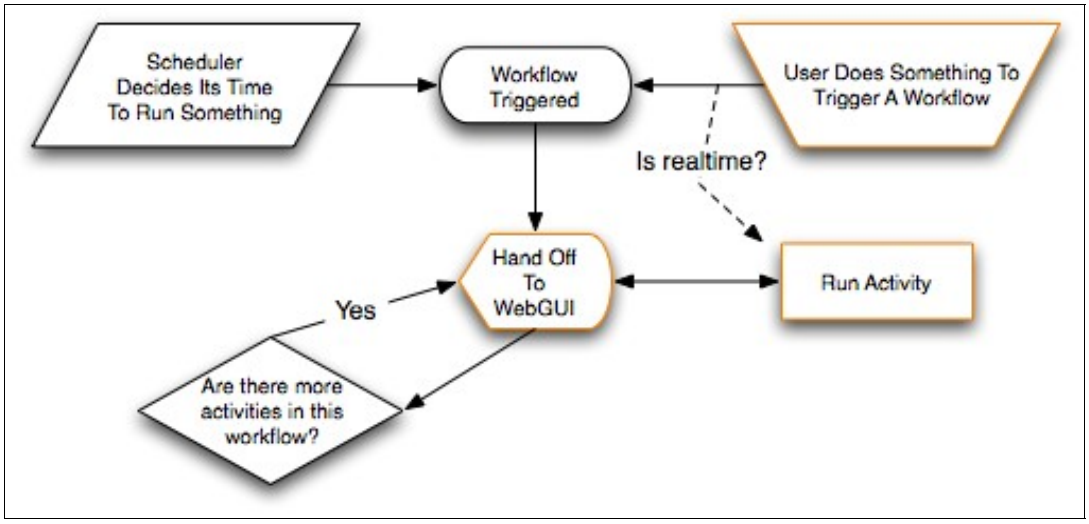
The Title field contains the name of this task. In this example, the Is Enabled? field is currently toggled to Yes, meaning that the task is enabled and the workflow will be performed at the scheduled time. If the Run Once? field is toggled to Yes then the event will run at its scheduled time then retire; however, if set to No it will run as often as scheduled. The Workflow field indicates the workflow that will be performed at the scheduled time. The Priority field allows you set the level of priority this task holds; if several tasks are scheduled to perform at the same time, the higher priority tasks will run first. The remaining fields are used to set the minute, hour, day, month and week this task will be performed. If an asterisk(*) is entered for the day field this task will run at the scheduled time every day. Likewise for the month and week fields.

To add a new task to the Scheduler, click on the *Add a new task* link on the far right hand side of the screen. A blank Scheduler screen will appear, with the same fields as the example above. Simply enter your settings and click save. Upon saving you will be returned to the main Scheduler screen where your new scheduled task will be added to the list.

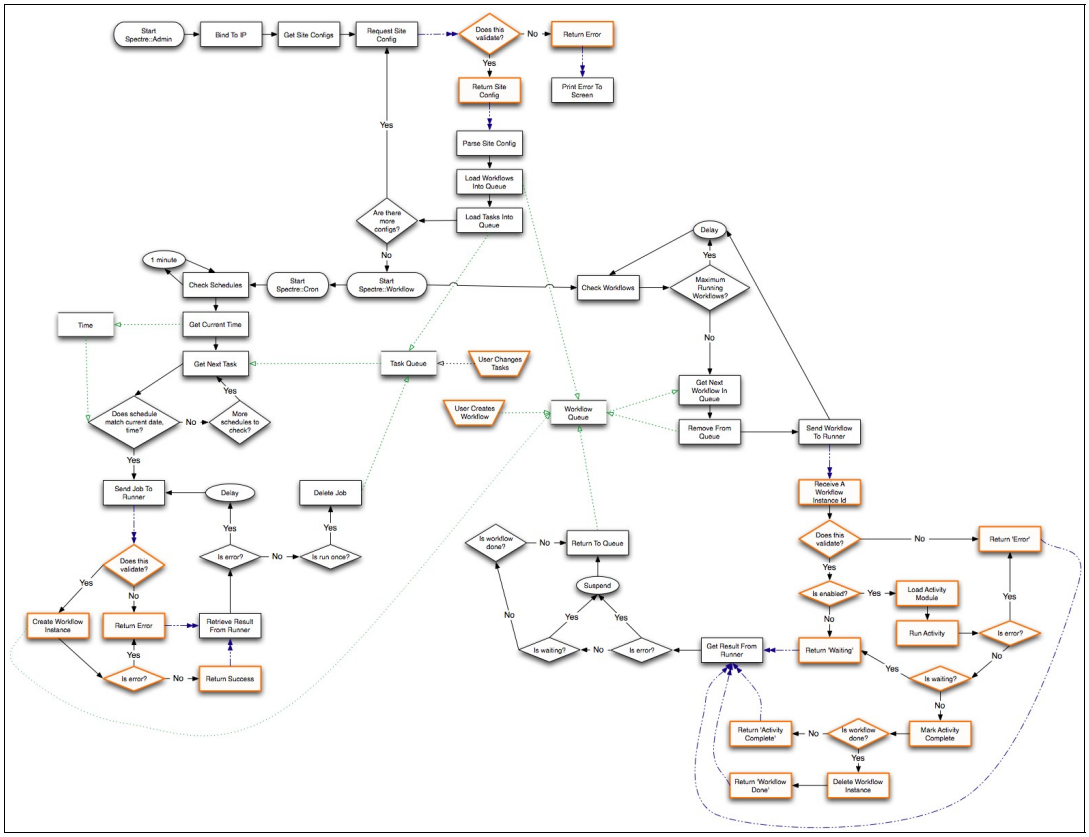
Workflow Engine

It's one thing to know how to use WebGUI Workflow, it's quite another to understand the engine that drives the whole thing. But if you know how it works, you can use it better, and troubleshoot problems better.

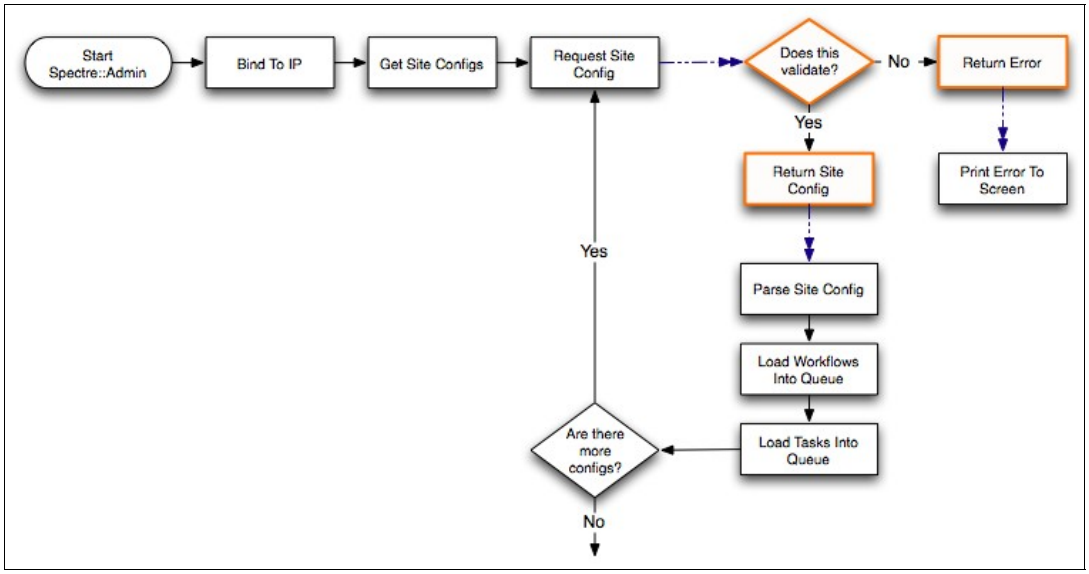
WebGUI's workflow engine is actually two systems, the first is WebGUI itself, the second is a governing application called Spectre. The diagram below shows a very high level view of this process. The components on the left are Spectre, and the components on the right are WebGUI.



Unfortunately, it's all a bit more complicated than that simple diagram. The next diagram shows the workflow engine in its complete detail. It's too big to read like this, so we'll break it down section by section as we proceed.



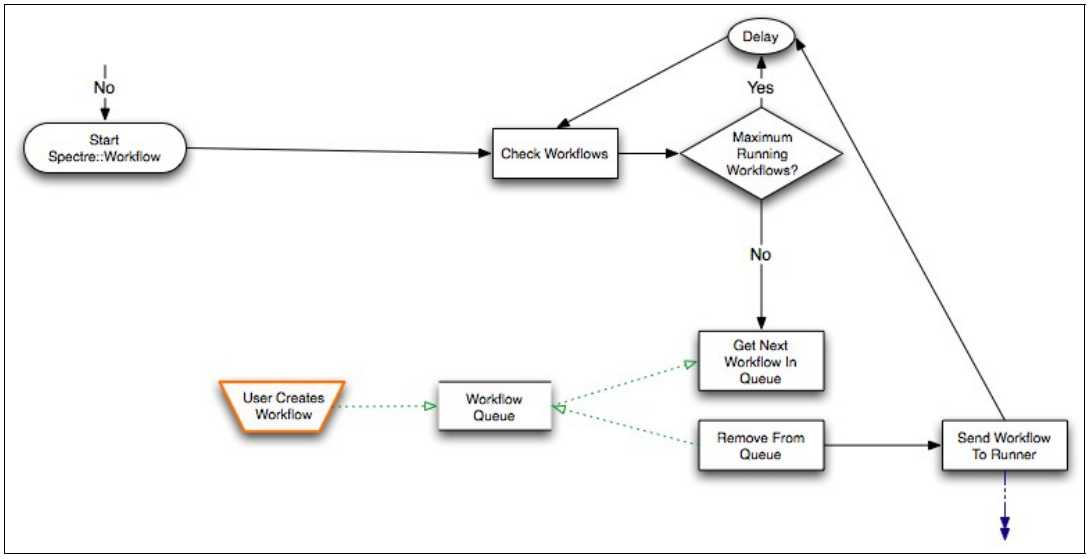
Let's start by looking at what happens when Spectre first starts up.



Spectre starts and binds to an IP address so that WebGUI can communicate with it. Then it loads the site configs, and continues loading them until there are no more to load.

Workflow Governor

Then Spectre starts the workflow governor, and starts checking to see if it has any workflows to run. Note that it also checks that it's not running more than the maximum number of workflows; it doesn't want to overwhelm WebGUI. When it determines that there is a workflow to run, and that there's room to run it, Spectre will send the workflow off to WebGUI to be run.



When WebGUI receives the request from Spectre, it first validates that the request is genuine and not from a hacker. It also checks to make sure that you, or another admin, have not disabled the workflow; there's no sense in running a workflow that's disabled. It will then load the next activity module in the workflow and run it. There are four possible outcomes from this:

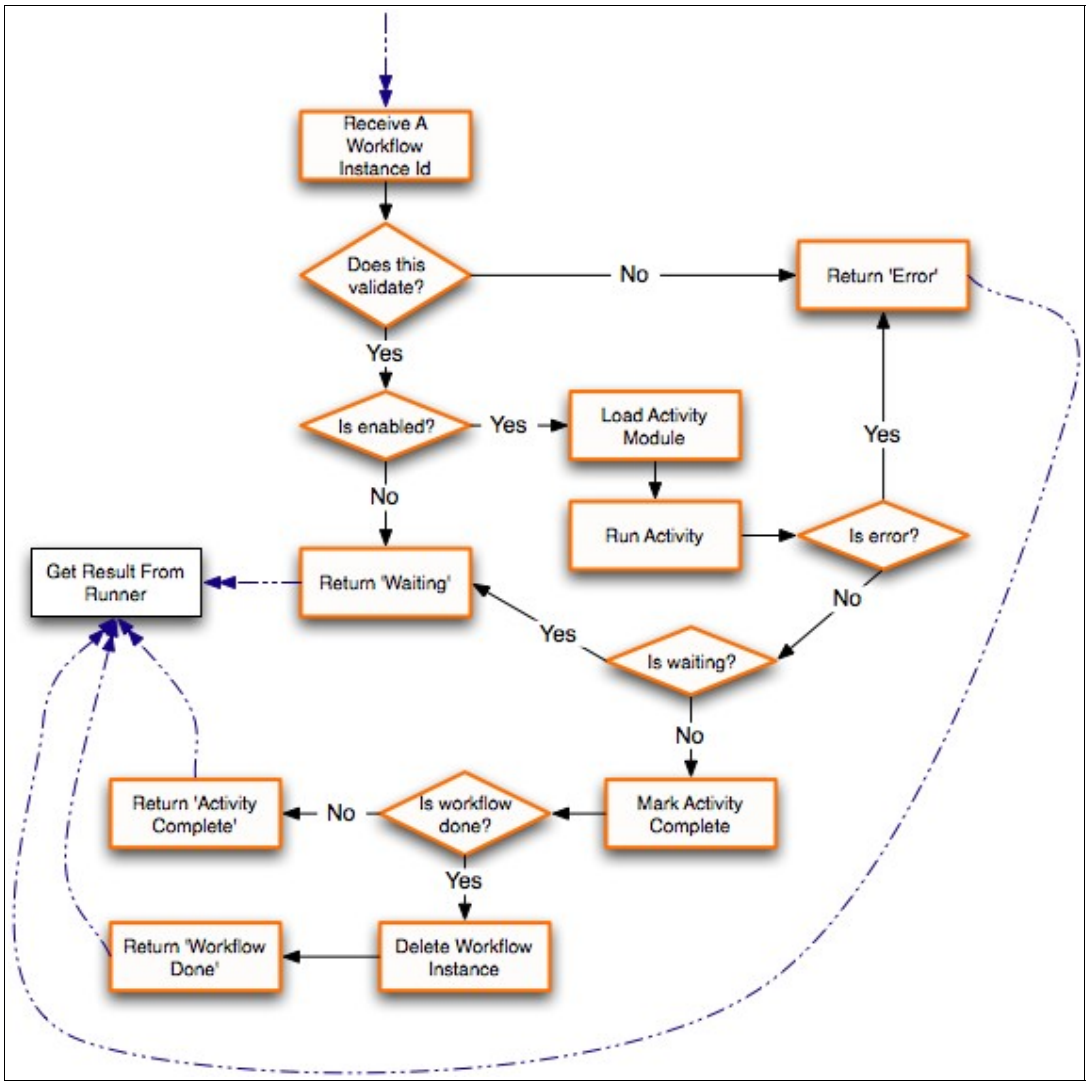
1. **done** – This means that the workflow has executed all of its activities successfully, and no longer needs to be run.

2. **complete –** This means that a single activity has completed successfully, and that the workflow governor may request that the next activity be run.

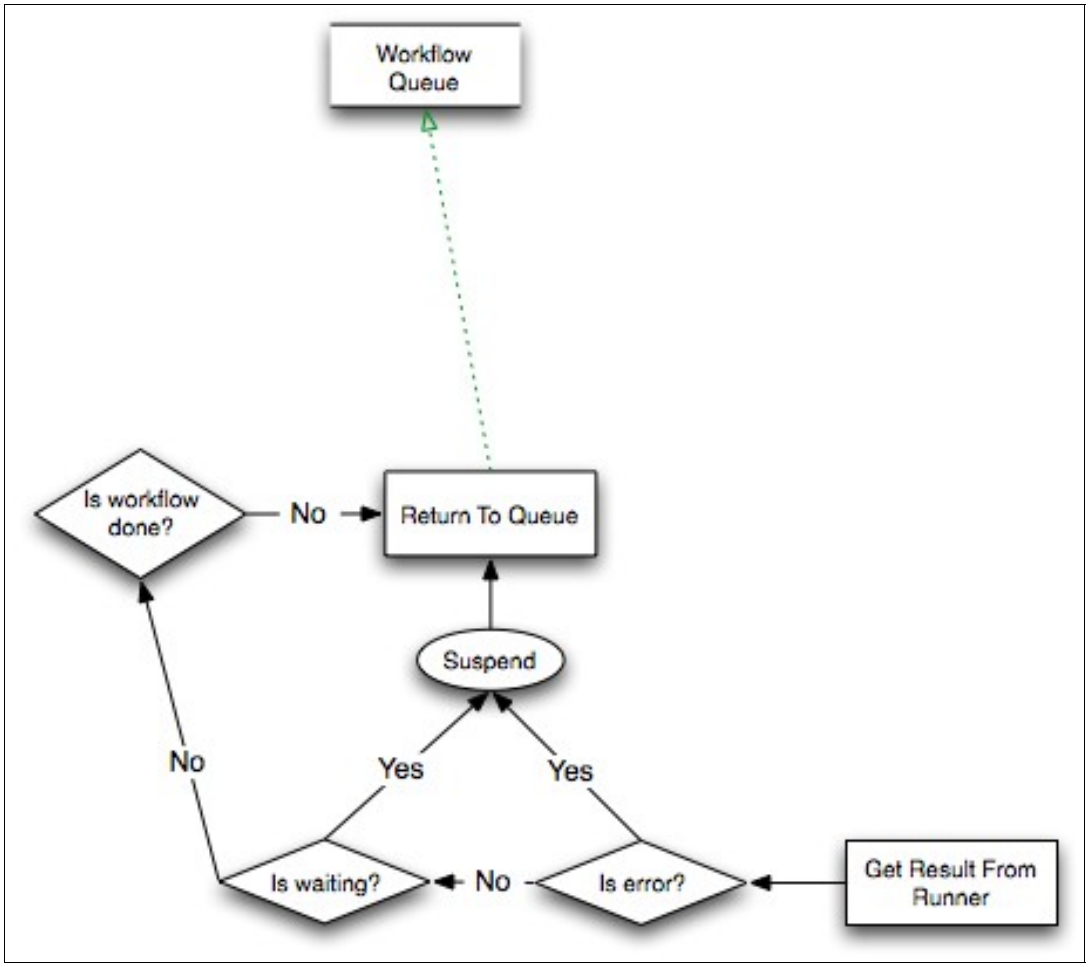
3. **waiting –** This means that the activity either ran out of time trying to

complete its work, or it's waiting on some external input; perhaps it's waiting on a user to approve some content. Either way, it tells the workflow governor that it needs to run this activity again later.

4. **error –** This means that the workflow activity could not complete its task. It could just be a network hiccup, but it also might be something more serious. The workflow governor will wait a while and try to run it again to see if the error goes away.

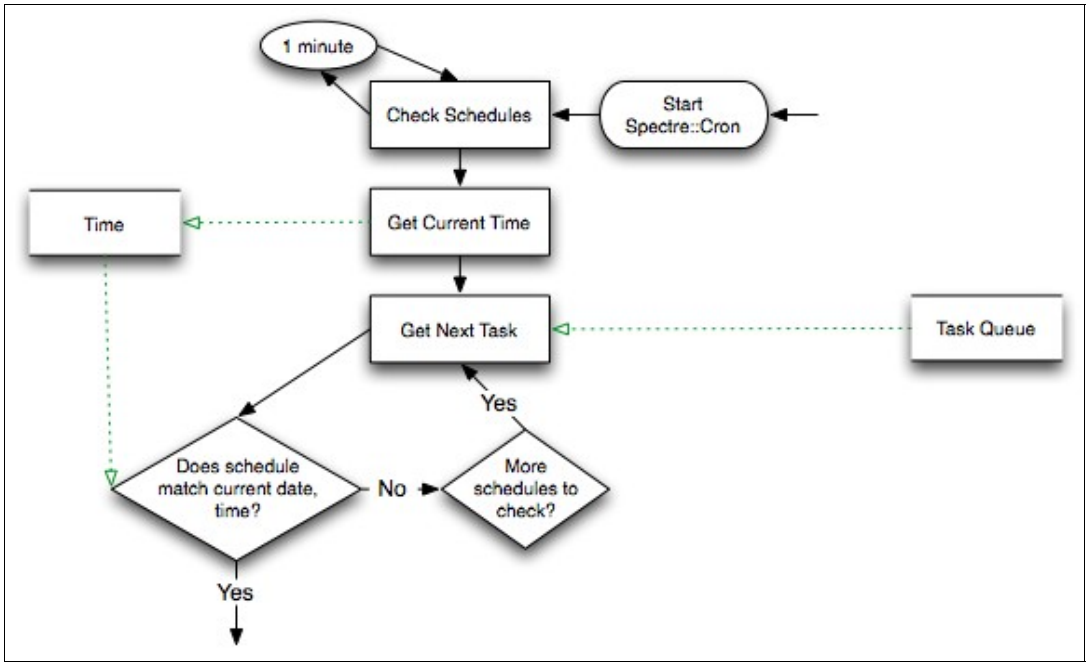


After execution is complete, WebGUI hands the result back to Spectre. Spectre doesn't return the workflow to its in-memory queue if the workflow is “done”, but otherwise it does so that the workflow can be executed again until its done.



Schedule Governor (Cron)

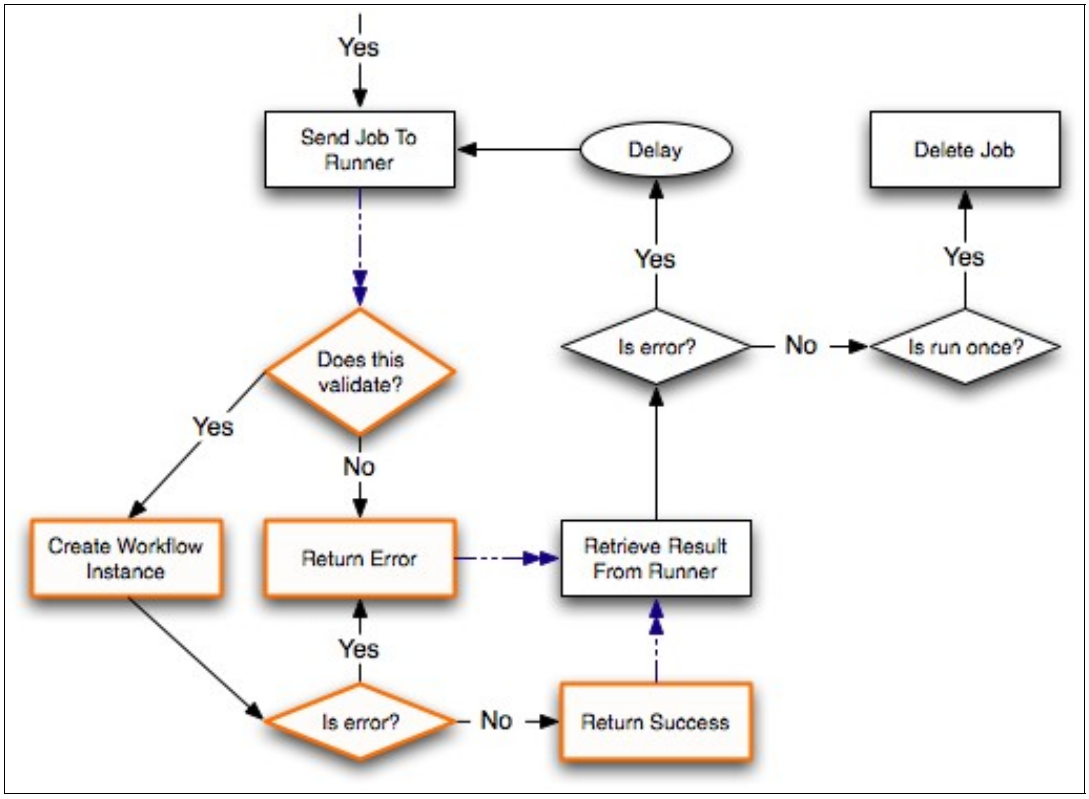
A very similar process to the workflow startup happens when the scheduler starts up. It begins checking its newly loaded schedules to see if any of them match the current time. It checks every schedule once per minute to find a match. If it finds a match, it then triggers a process to create a new workflow instance by contacting WebGUI.



Once the scheduler has found a task that's ready to run, it sends it off to WebGUI, where WebGUI does the same validation process it does when the workflow governor sends over a workflow to execute. WebGUI then attempts to create a new workflow activity based upon the data it has. It then can return one of two status codes:

1. **success –** When the workflow is created.

2. **error** – When something bad happens. This can only really happen under two circumstances. The first is if there was some sort of network connectivity problem. The second is if there is some configuration problem or data corruption with the task and the workflow instance it's supposed to create.


Yes
Send Job To Runner
Delay
Delete Job
Yes
Yes
Does this validate?
Is error?
No
Is run once?
Yes
No
Create Workflow Instance
Return Error
Retrieve Result From Runner
Yes
Is error?
No
Return Success

Versioning

The versioning system in WebGUI versions the properties of assets so that you can easily undo changes that you or your users mistakenly made. It also allows you to make changes to your site and not publish them until you are satisfied that they are ready to go out.

Revisions and Versions

A revision is a change to a particular asset. A version tag (or a version) is a collection of one or more revisions. A single version tag may include multiple revisions of the same asset, or multiple revisions made to more than one asset.

WebGUI works in terms of version tags rather than revisions, because often you'll want to make several changes before publishing them live on your site. However, if you prefer to have the option, you can change the "enableSaveAndCommit" option in your WebGUI config file to enable a second button on all your assets.

"enableSaveAndCommit" : 1,

You'll then see buttons that look like this when editing assets:

What Is Versioned?

You might think that since there is a versioning system that all data would be versioned. However, not all data needs to be versioned, or is practical to version. For this reason only asset data is versioned, not data like users, groups, and profiling data.

Within assets, only the properties of an asset and the files attached to the asset are versioned. Assets like Data Form and Survey have collateral data like fields, questions, and answers. These collateral data fields are not versioned unless the asset developer specifically decides to version them.

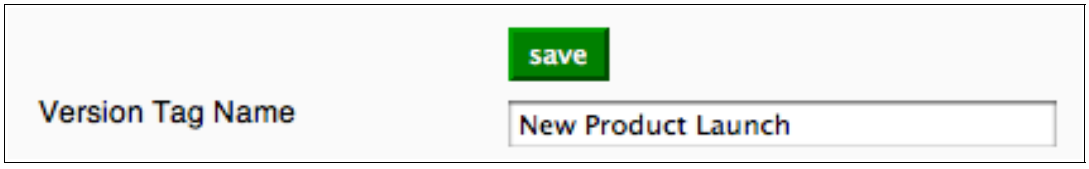
Finally, the relationship of assets to each other are not versioned; therefore, cut, paste, and delete functions are not versioned. If you cut a page from

one place, and paste it to another place in the site, that change will be reflected immediately on the site.

The User Interface

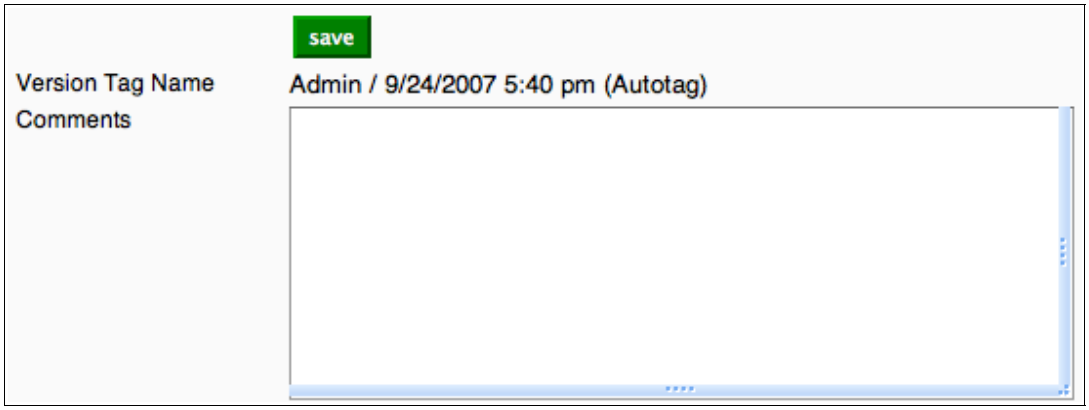
To use the versioning system you just need to edit something. A version tag will automatically be created for you. When this happens you'll have a new tab in the admin bar that looks like this:

If there are other tags open you can click on one of them to join it. Two people can be working under the same tag at the same time. In this way you can collaboratively develop content. Click on the highlighted version tag again and you'll be taken to the screen to edit the version tag. Here you can rename the version tag. Naming version tags is a good idea if multiple version tags are present on the site. This helps avoid confusion over what is being edited under each tag.



Committing A Tag

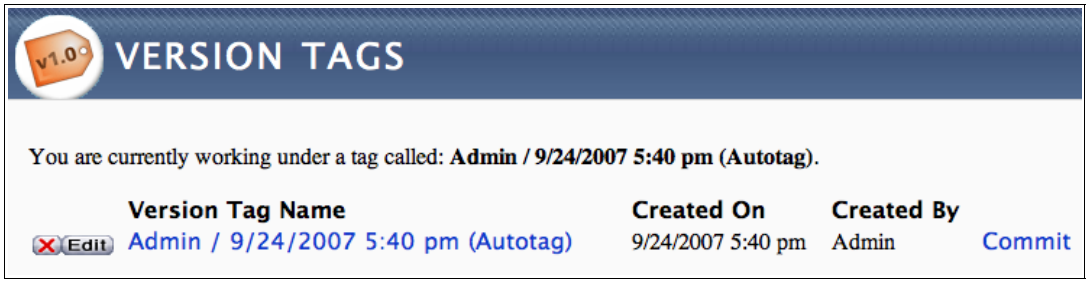
If, instead, you click “Commit My Changes” you'll be taken to a screen where you can attach comments about the changes you made. These can be invaluable if you need to look back in the future to find out why a particular change was made.



Open Tags

The admin bar isn't the only place to see the open version tags. While making further edits to assets you'll see all the open version tags in the menu on the right. side of the screen, which looks like this.

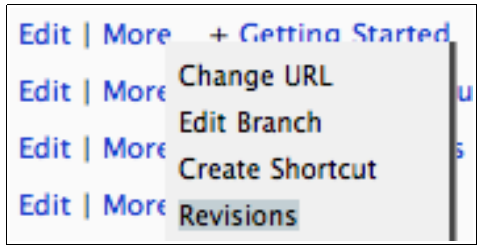
You can also go to Admin Console > Version Tags.



Viewing Revisions

You can look at all existing revisions of a particular asset. While editing an asset the top five edits will be available on the right menu, and will look like this:

If you click on the word “Revisions” you'll see a list of all the revisions of this asset ever made. Or, if you click on an individual revision's time the edit interface will actually display what the asset properties were with that edit. You can also get the the revisions list from the “More” menu in the asset manager, or the class icon in the inline view.



Locking

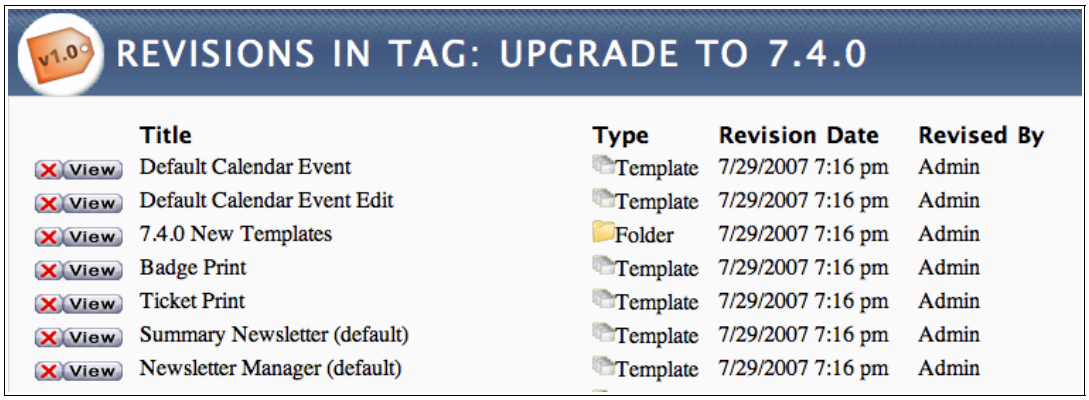
Once you have edited an asset, it becomes locked for someone else to edit unless they are part of your version tag, you commit your version tag, or you delete the revision of that asset in your version tag. In the toolbar they'll see a lock icon instead of the delete and edit icons, like this:

If you know you want to edit an asset, but you're worried that someone else might also want to edit it, and therefore lock it under their version tag, you can use the lock feature, under the class icon. This just creates a new revision of the asset with the exact same properties as the old asset, thereby including it in your version tag.

Revisions In A Tag

Under Admin Console > Version Tags you can not only see all the open version tags, but you can also see all the version tags that have been committed. There is a menu on the right side of the screen to “Manage Committed Versions”.

Whichever you are looking at, you can click on the tag name in the list to display a list of revisions in the tag. It shows you the asset title, type, who revised it and when. You also have delete and view options. If you delete the revision then that asset will revert to its old revision back on the site. However, if you delete the last revision of an asset, that asset will cease to exist entirely, so be careful. If you click on the view button you can see how that asset looks as of that revision.



Restoring An Old Version

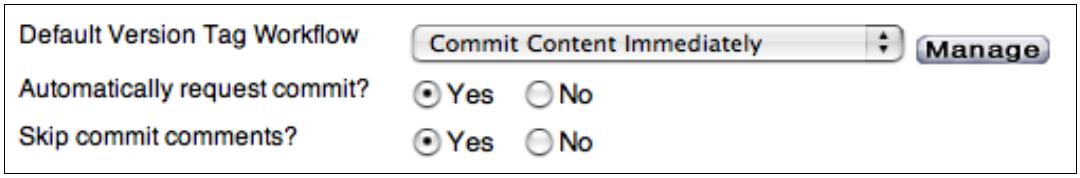
If something should happen and you no longer want a version tag around, you can do a “Rollback” on it. What that means is that all the changes of that version tag will be eliminated as if they never existed. Therefore, all assets will revert back to their previous versions, or if they were new in this version tag, then they will cease to exist entirely. This is usually not something you ever want to do, unless you wanted something to appear on your site temporarily, or someone has made a big mistake.

To use this function, go to Admin Console > Version Tags and then choose “Manage committed versions” from the right menu. You'll then be presented with a screen that looks like this:

From that screen choose the “Rollback” link on the item you wish to permanently remove.

Hiding Versioning And Workflow

Versioning and workflow is a very powerful combination, but often on small sites it's just overkill. The good news is that you can hide it from your users to make the publishing process easier. Go to Admin Console > Settings. Then click on the “Content” tab. Set the “Default Version Tag Workflow” to “Commit Content Immediately”. Then set “Automatically request commit?” and “Skip commit comments?” to “Yes”. Voila, your users won't know that workflow and versioning even exist.



In The Database

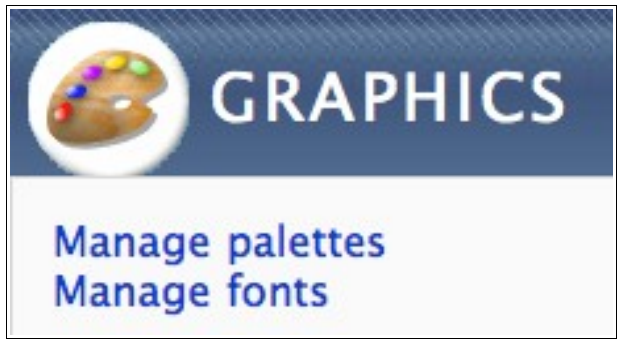
In the database a revision is nothing more than a database table row, or rather a series of rows, one for each table related to the asset. Take Article for example. It is made up of the asset, assetData, wobject, and Article tables. The asset table contains the information that isn't versioned, such as the article's position in the asset tree. The other three tables each have an assetId field (which uniquely identifies this asset), and a revisionDate field, which uniquely identifies a revision within an asset.

To find out what version tag a particular asset revision belongs to, look for the tagId column in the assetData table. The tagId column relates to a row (the tag) in the assetVersionTag table.

As an administrator you should never have to know this. It's more the domain of a programmer. However, if you're an administrator that also can do a little coding, this information can be invaluable with reporting and troubleshooting.

Graphics

WebGUI has a basic graphics subsystem which can be used to draw things like charts and graphs. In the future it will be able to do much more, including photo manipulation.

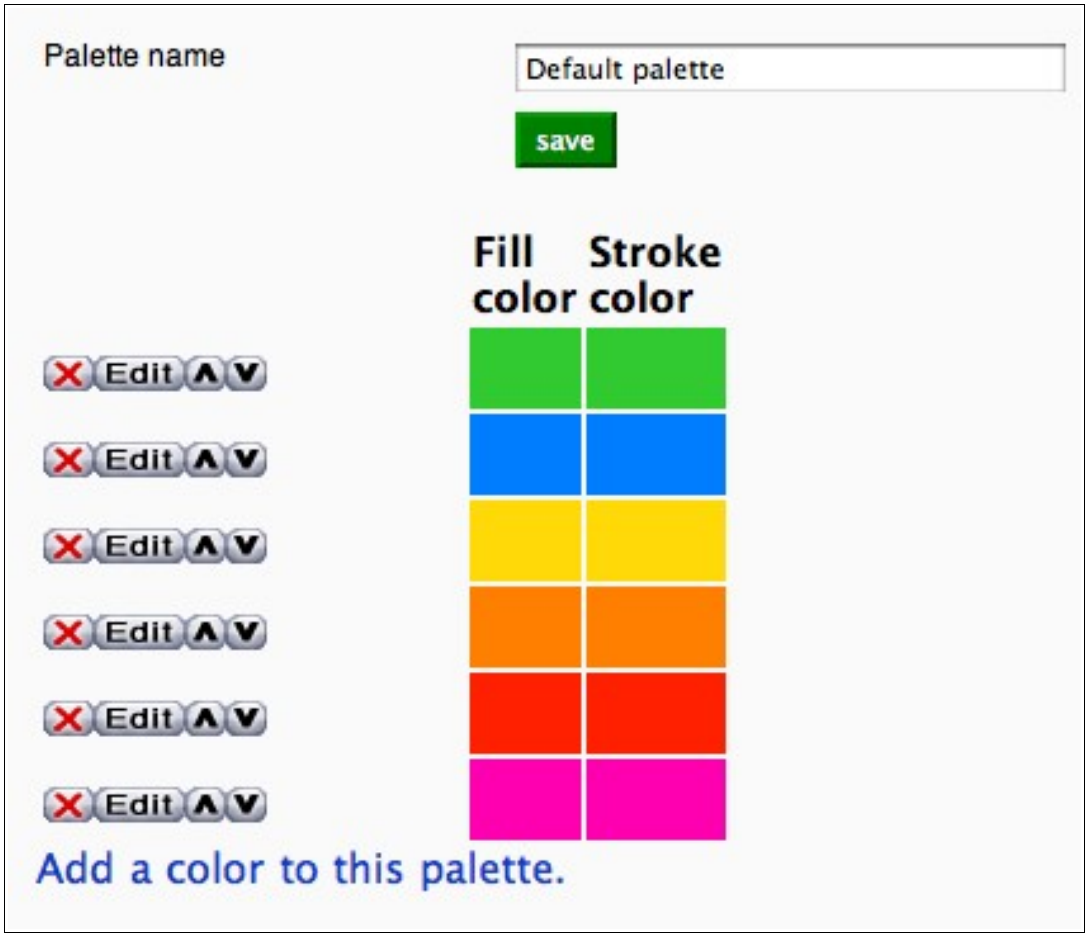


Under Admin Console > Graphics you'll see a screen like the one above. From there you can manage either palettes or fonts.

Palettes

If you click on “Manage palettes” you'll see a screen like the following:

A palette is a grouping of colors. In WebGUI, palettes are used to choose what colors go into a chart, like the charts you find in the Poll asset. The default palette is a bunch of bright colors like this:



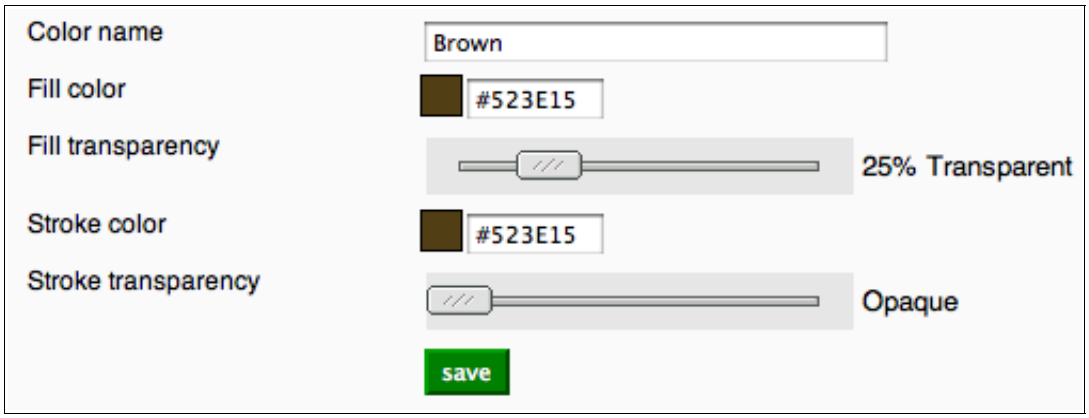
However, you can easily define your own palette to match the colors on your site. Let's say your site is about how you like autumn and the colors changing on the trees at that time. Any polls you create you'd of course want to match that color palette.

1. Start by clicking “Add a new palette” from the manage palettes screen and give your palette a name.

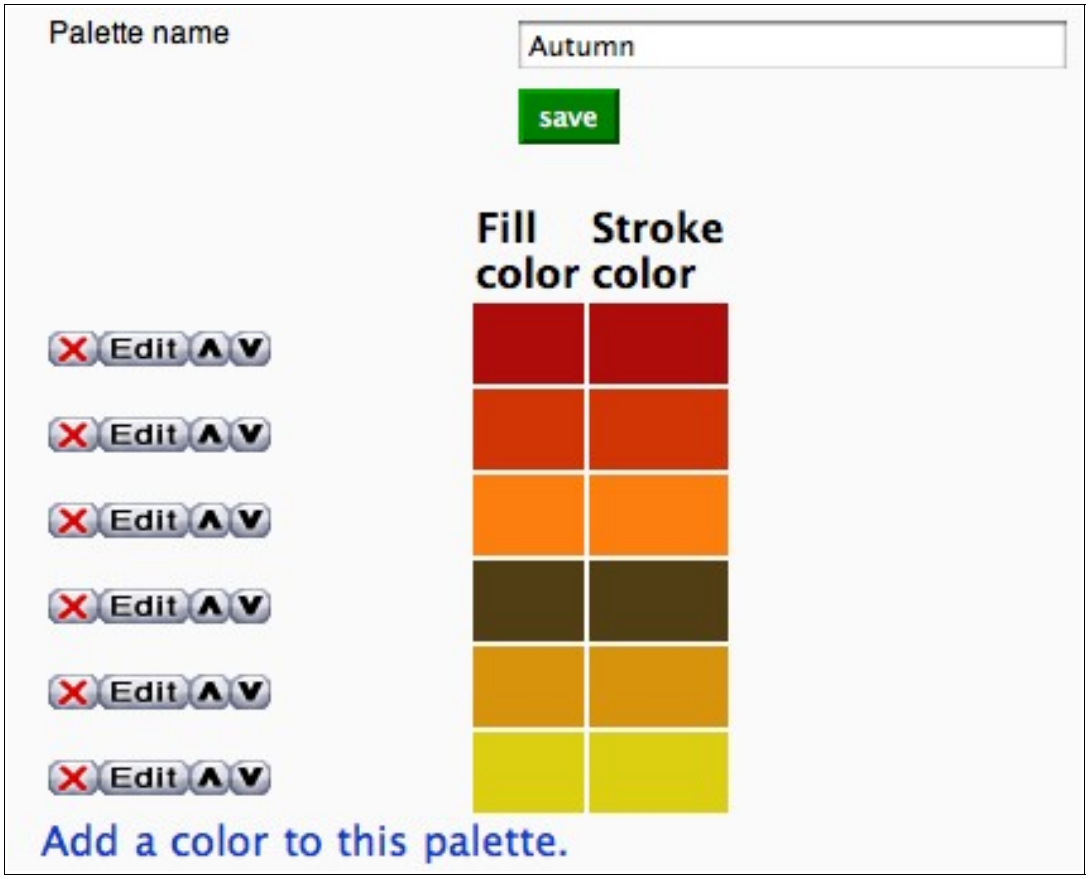
2. Now, you can add colors to your palette. Give the color a name.
3. Then, choose the fill color. The fill color is the primary color of the

surface of the graph element. Along with the color you can pick the transparency of that color, all the way from opaque to completely transparent. Transparency can provide a neat effect, particularly in 3D graphs.

4. Now you can choose the stroke color. The stroke color is the color the borders or outline of the chart element will be drawn in. We recommend making this either completely opaque, or slightly less transparent than the color itself. If the color is completely opaque, then choose a color that has the same hue, but is slightly darker. It really makes the drawn elements pop off the screen.



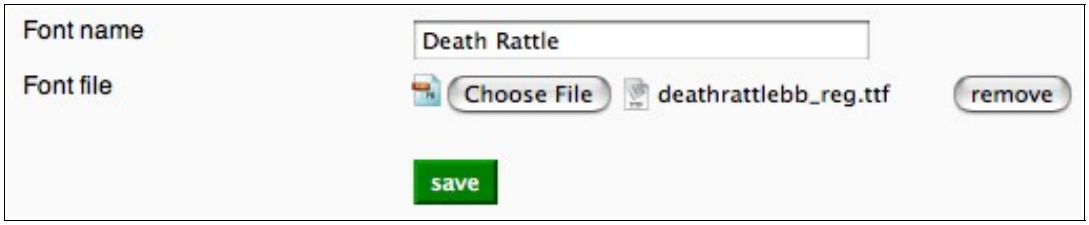
5. Keep filling in colors until you have a palette of six to eight colors.



Fonts

Fonts are used when drawing text onto an image. WebGUI comes with one built in font, which is a very basic legible, but otherwise bland, san-serif font.

You can add your own fonts for a little extra pizzazz. Click on “Add a new font”. Then give the font a name. This is what will appear in forms when you choose a font. Then, specify a font from your hard drive. WebGUI supports TrueType® and FreeType fonts.



End Result

It takes a bit of effort to pick all your colors and choose just the right font, but when you're done, it's well worth it because you can create exactly what you want.

Advertising

The advertising module in WebGUI gives you an easy way to put advertisements into a rotation on your web site. You may be thinking to yourself, “Why is that useful? I get the code from my ad partner and put it into my HTML. I don't need no stinking advertising system.” You're only half right if that's what you're thinking. Depending upon what your site is about, you may be able to make more money by selling ads directly to advertisers. In addition, you may be able to make more money by using more than one ad network. Using the ad system allows you to distribute ads for multiple ad networks in the same space, and allows you to throw in your own advertisements as well.

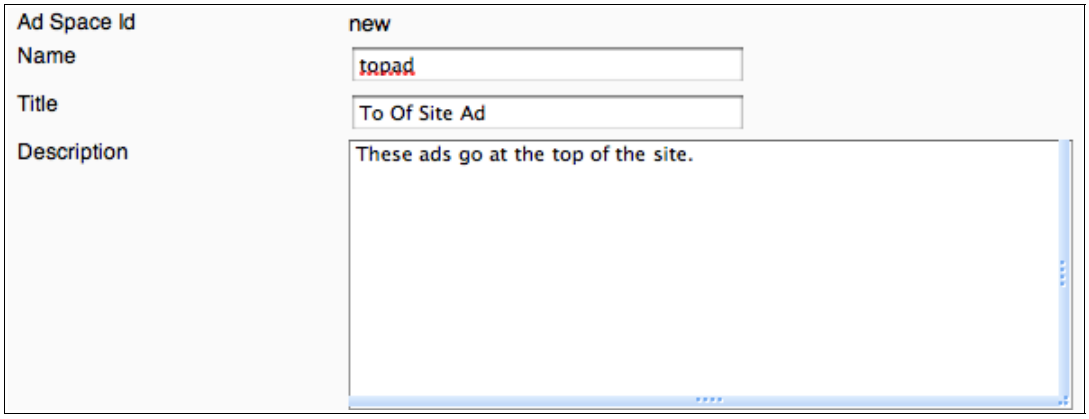
Beyond selling advertising on your site, have you ever thought about how you could use the popular areas of your site to increase awareness of the newer or less frequented areas of your site? You can do this, too, using the ad management system. Instead of placing advertisements, you're placing your own site elements into the rotation. You can see this in use on the front page of plainblack.com:

Creating Ad Spaces

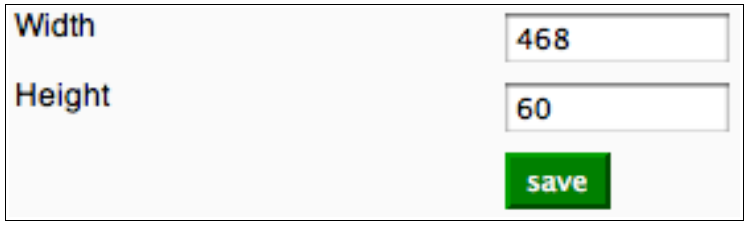
An Ad space is a placeholder for where you will put advertisements. With it you can define the size of ads allowed, and where they are placed on your site.

1. To create an ad space go to Admin Console > Advertising. Then choose “Add ad space.” on the right side.

2. Give a name, title and description to your ad space. These can each be anything you want them to be, but the name should be unique amongst all ad spaces defined on your system.



3. You can also define how large of an ad space this will be. The ad system will enforce this so that ads can't disrupt the look and feel of your site even if someone adds an ad that is too big for the ad space.



4. Hit save when you're done editing this new ad space. You'll now see a box appear that shows you how to use the ad space.

You'll note that an ad space is placed using nothing more than a macro. It takes the form of ^AdSpace(ad space name);. You can define as many ad spaces as you want on a site, and place as many as you want into a given page. Normally you'd place the AdSpace into a style template.

Create Ads

Now that you have an ad space, create an ad.

1. From the ad space screen click on “Add an ad.”

2. Enable the ad. You can disable the ad if you ever want to stop running it, but don't wish to delete it.

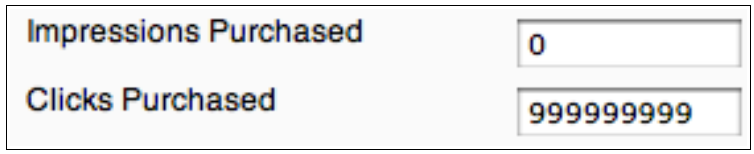
3. Then, fill in the title. The title is displayed in text ads, and is used as the alt text in image based ads.

4. Now, fill in the URL. This is where the user will be directed to if they click on text based or image based ads.

5. If you wish, you can set the priority of the ad. The closer the priority is to 0 the more often the ad will be displayed. This is not an exact ratio, but rather a scaling mechanism. The more impressions per minute your site displays the larger the impact of the scale. If your site only displays one impression per minute, then priority will have no effect.

6. Now, enter the number of impressions or clicks to be displayed for this ad. The ad will only be displayed that number of clicks or

impressions. If you want the ad to run indefinitely then enter 999999999 clicks. Note that you can only use impressions as a counter for rich media ads.



7. Finally, choose the type of ad that you wish to display. You can choose from text, image, and rich media ads.

Text Ads

Text based ads generate the ad using just text and a simple color palette. First enter the text of the add, which will appear under the title. Then pick the colors you'd like the ad to use.

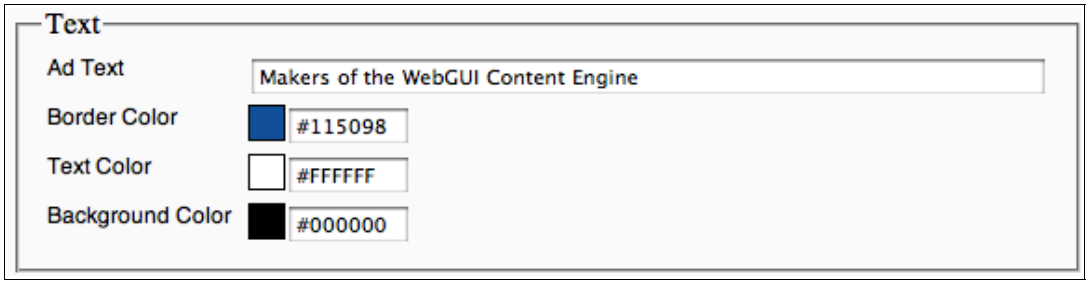
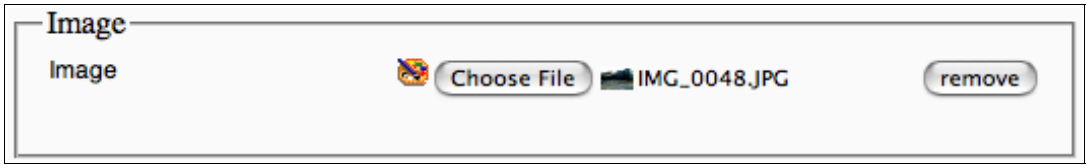


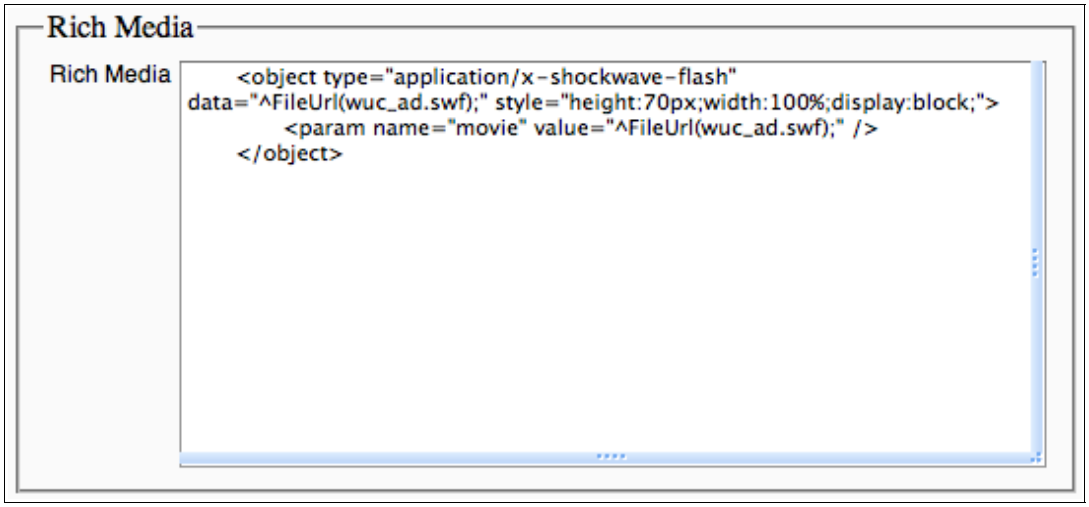
Image Ads

Image based ads are the more traditional banner ads that you see around the world wide web. If you selected image based ad then you can upload an image to use here.



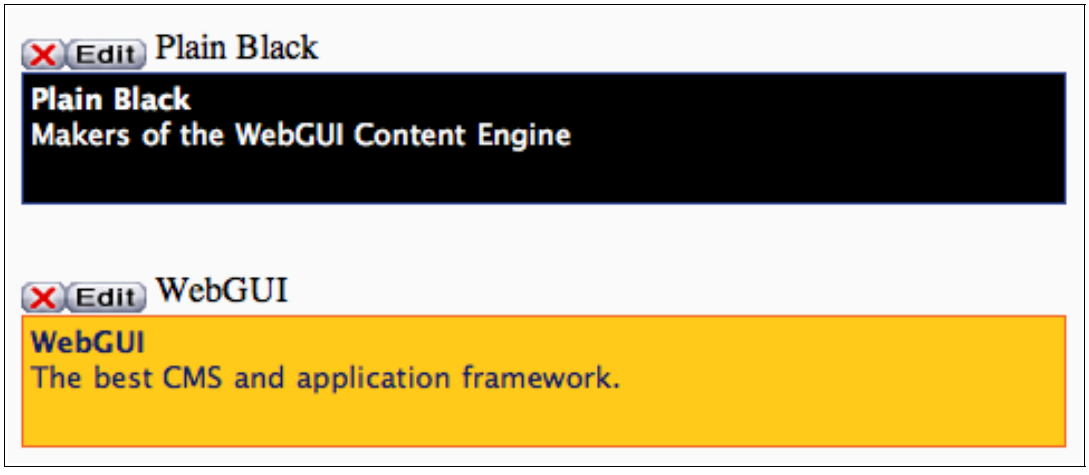
Rich Media Ads

Rich Media ads are ads that allow you to place HTML tags in them. Using this you can make ads from complex HTML, video files, or flash movies. Simply place the HTML into the rich media field.

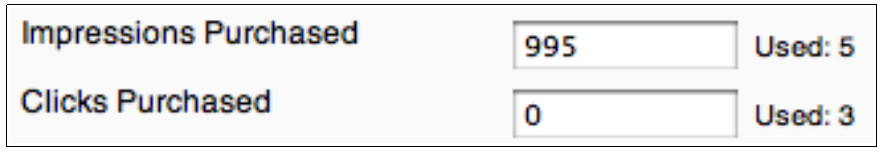


Viewing Ads

If you edit an ad space after you've defined ads you'll see a list of the ads assigned to this ad space.



If you edit one of those ads, you'll be able to see how many impressions and clicks it's had.

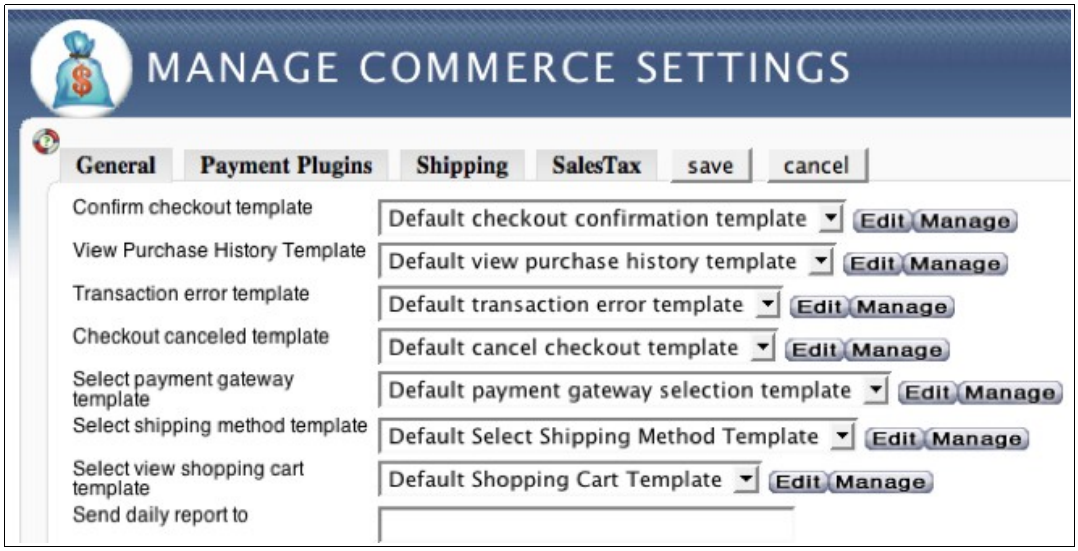

Impressions Purchased
995
Used: 5
Clicks Purchased
0
Used: 3

Commerce

To stay competitive in today's marketplace many businesses are offering their products and services for sale on their websites. Others use online commerce to sell subscription services to clients. WebGUI comes with tools to handle all commerce needs over a secure connection.

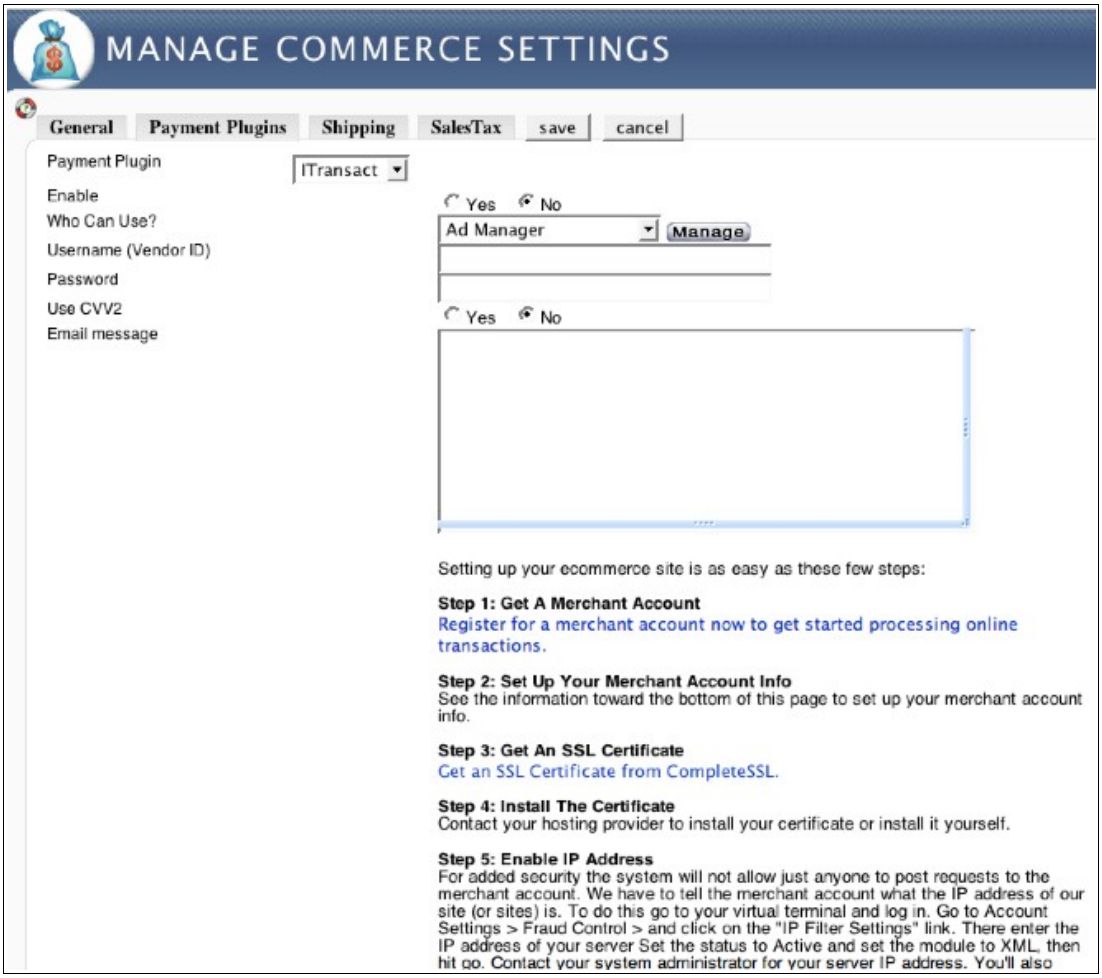
To sell something on a WebGUI site, you must first set up the payment gateway.

1. In the Admin Console, click on the Commerce link to open the "Manage Commerce Settings" screen.



The **General Tab** of the Manage Commerce Settings screen contains templates for the display of your commerce system. The default templates will automatically be displayed. To manually edit these templates, click the template's edit button.

2. Click on the **Payment Plugins Tab**. This screen displays a list of instructions. Simply follow these instructions and choose the appropriate links to set up your site for online transactions.



3. **Step 1** on this screen is Get a Merchant Account. If you still need to set up a merchant account, click on the link under this step. You will be directed to the CDG Commerce© site where you will be instructed to complete an online form. Follow the directions on this site to complete your merchant account acquisition.

4. After you have acquired your merchant account you need to set up your merchant account information, as instructed in **Step 2**. There are instructions near the bottom of this screen providing further information. Please carefully follow these instructions.

5. **Step 3** instructs you to get an SSL Certificate to ensure the security of all your online transactions. Clicking on the link below Step 3 will direct you to the Complete SSL© website. Plain Black and Complete SSL have teamed up to pass along savings to you, and there a

number of options for you to choose from. Simply make your selection and follow the instructions given on the Complete SSL site to purchase your SSL Certificate.

6. **Step 4:** Once you have purchased your SSL Certificate, you will need to either contact your hosting provider to install it, or install it yourself if you know how.

7. **Step 5** is to enable the IP Address. For added security, you must tell the merchant account the IP Address of your site(s). There are detailed instructions on the screen to guide you through this.

8. Once you have acquired and set up your merchant account and SSL certificate, you need to enable the commerce system on your site as directed in **Step 6**. To do so, simply set the Enable field, located at the top of this screen, to Yes.

9. **Steps 7** and **8** provide instructions on how to accept alternative forms of payment. By default, you will be able to accept Mastercard and Visa; if you wish to accept other forms of credit card payment, or if you wish to accept eChecks, you will need to follow the instructions and link provided in Steps 7 and 8 to set this up.

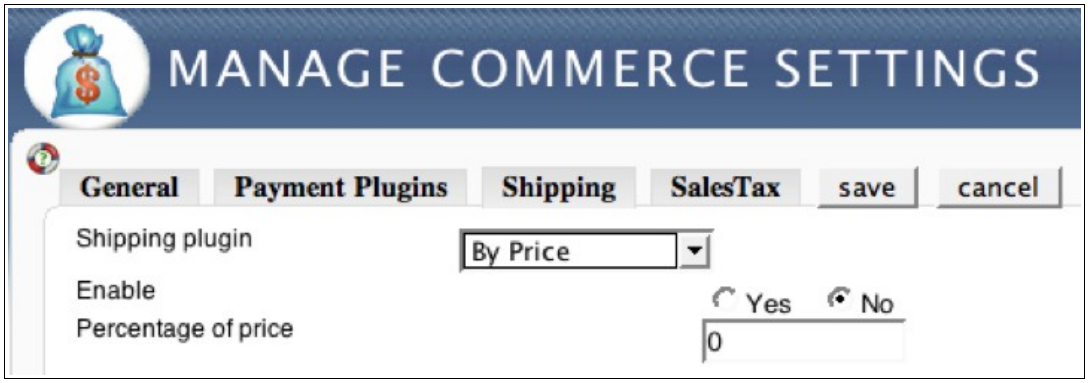
10. To complete your setup in this screen:

    - Enter your username (vendor id) in the “Username” field. This was established at the time the merchant account was created.

    - Enter your password in the “Password” field.

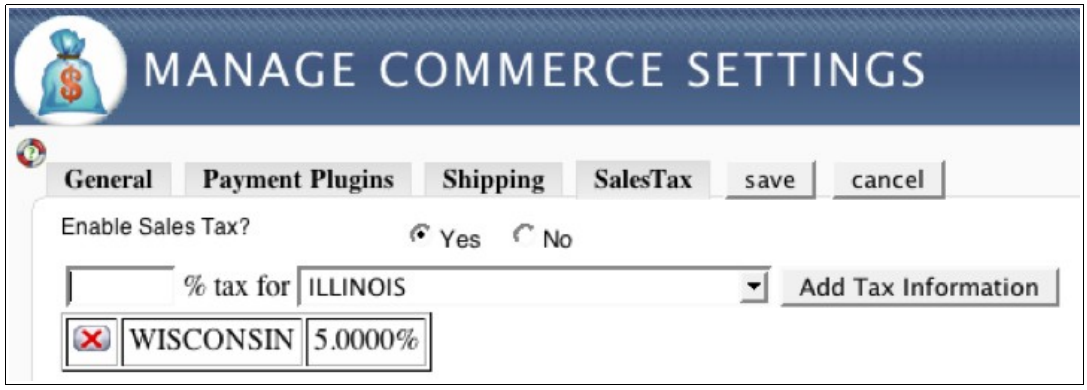
    - Click save, and your WebGUI site is ready to begin processing online transactions.

Shipping Tab



The “Shipping” tab allows you to manage shipping costs. In the “Shipping plugin” dropdown menu you may choose to charge shipping costs based on price, transaction, or weight. In order for the shipping cost to be applied to transactions, you must also set the “Enable” field to Yes, and indicate the price to be applied to the shipping cost in the “Percentage of price” field. The wording of this field will vary depending on the selection made in the “Shipping plugin” field.

Sales Tax Tab



The Sales Tax tab allows you to enter the sales tax for your state. Set the “Enable Sales Tax?” field to Yes, enter the percentage of sales tax for a chosen state, then click the “Add Tax Information” button. The state’s sales tax will then appear at the bottom of the screen; to remove a state click its red X icon.

Add Products to Commerce System

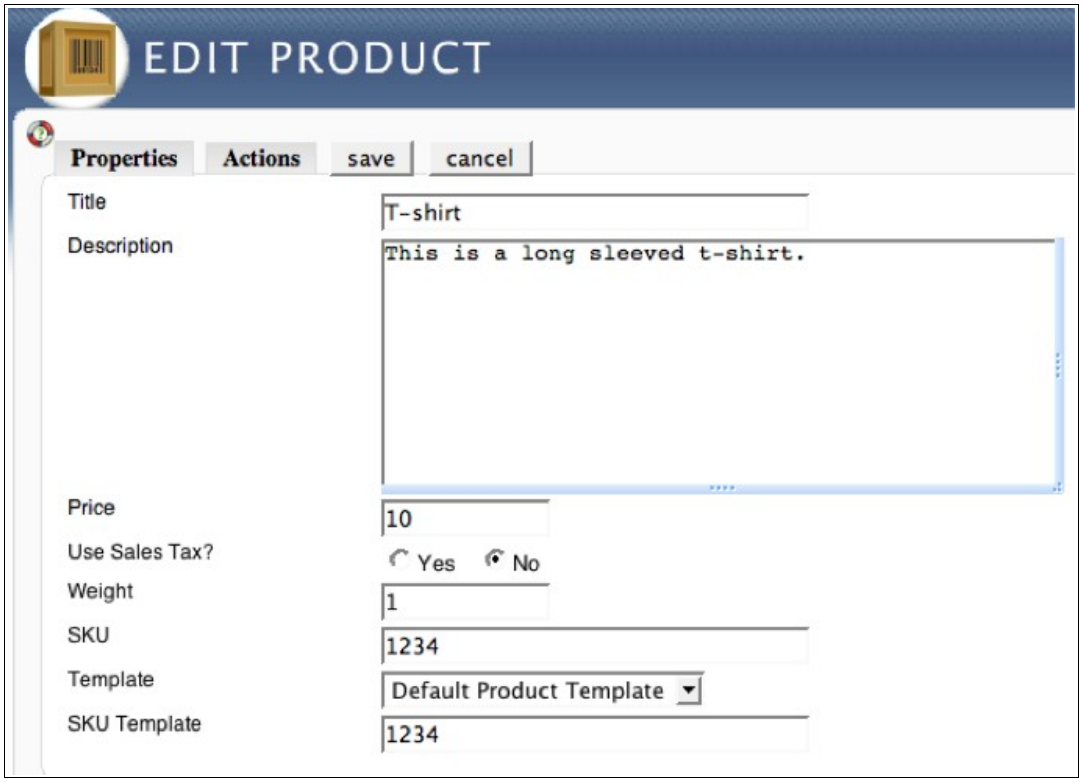
To add merchandise or services for sale, use the Asset Manager. Click on the Assets link in the Admin Console to open the Asset Manager.

Open the Admin Console and click on the Products icon. The “List Products” screen will open. From this screen you can add products to sell on your website. If you already had products on your site for sale, they would appear here as well.

To add products, click on the “Add a new product” link at the far right hand corner of the screen. The “Edit Product” screen will open. On this screen you can enter product information as well as set a limit of time the product will be displayed for sale on the site.

Properties Tab

In the Properties tab you can enter a description of your product, as well as indicate the price and assign a SKU.



1. Enter the title of the product in the “Title” field.

2. Enter a description of the product in the “Description” field.

3. Enter the price of the product in the “Price” field.

4. If you chose to apply sales tax when you set up your commerce system, and you want that sales tax applied to this product, toggle the “Use Sales Tax?” field to Yes.

5. Enter a SKU number in the “SKU” field. This may be whatever value you wish to enter.

6. Leave the “Template” field set to the default, unless you want to customize a template of your own.

7. Enter a “SKU Template.” You maywant to make this identical to your SKU number. Template variables can beused in this field to dynamically generate SKU's. This can also be done later in the process.

Actions Tab

The “Add to Group” field indicates a field users will be automatically added to upon purchasing this project. If this is a subscription purchase, select the group allowed to purchase it (this group must first exist in WebGUI. See chapter on Users and Groups for more information.)

The “Group Expires Offset” field indicates the amount of time to pass before users who purchase this product would be removed from the group in the “Add to Group” field. Users would have to purchase this product again to be added to the group once this amount of time has passed.

When you’ve completed the fields in the Edit Product screen click save. You will be directed to the Manage Product screen.

Parameters

You will be directed to the Manage Products screen by default after adding a new product from the List Products screen.

If you have different types of the same product to sell, you’ll want to add parameters to differentiate between those types. In this example, t-shirts are being sold; however, so far no differentiation has been made as to whether the t-shirt is red, blue, green etc. Parameters are added from the Manage Product screen. The Manage Product screen is accessed through List Products screen. Go to the Admin Console and select Products. All the products for sale on your site will be listed.

To access the Manage Products screen from the List Products screen, click on the Edit tab to the left of the product you’d like to manage. This will open the Manage Product screen, from which you can add parameters,

such as colors or sizes, to the product.

Add a Parameter to the Product:

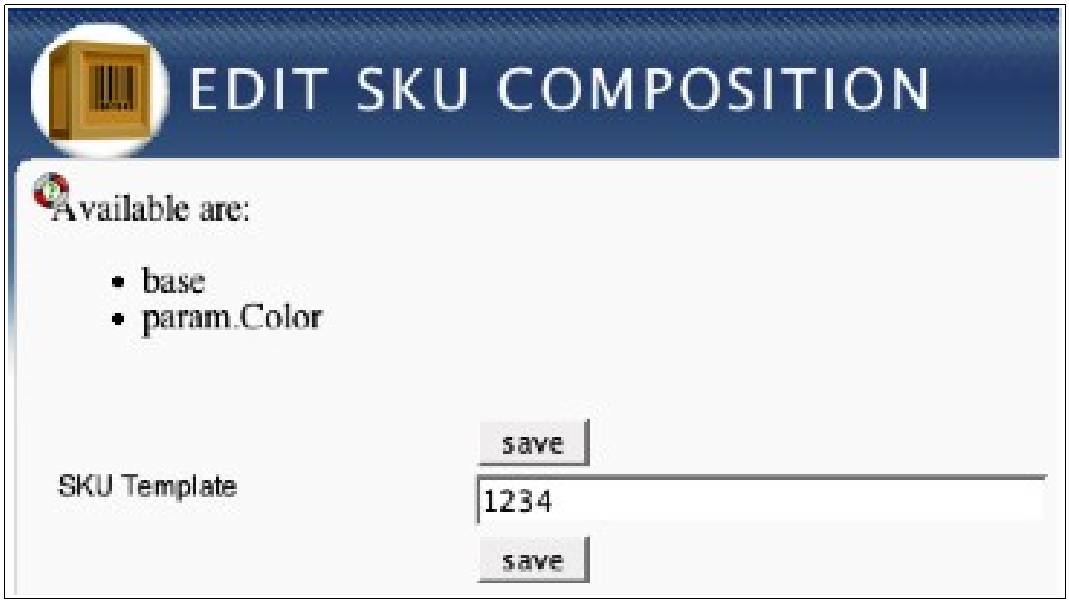
1. Click on the “Add parameter” link below your product properties.

2. The “Edit Product” screen will open.

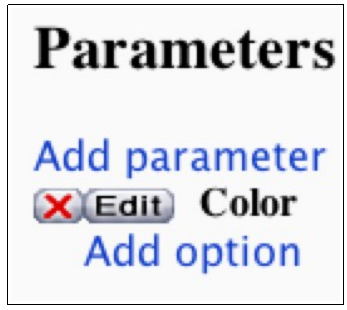
3. Enter the type of parameter you are creating in the “Name” field. In this example different colors of the t-shirt will be created.

4. Click save.

5. A screen displaying the SKU template will be shown. On this screen might also be displayed template variables. If you wish to use these template variables, enter them in the SKU Template field. Click save.



6. The new parameter (color) will appear under the Parameters field on the Manage Products screen.

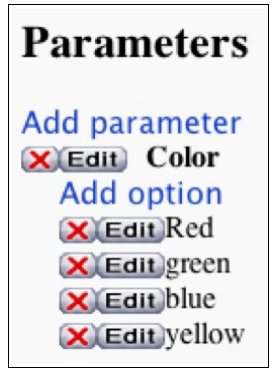


Now you can add options under the color parameter. (In this example, different colors in which the t-shirt is sold.) Click on the “Add Option” link under the Color parameter to enter the color options.

1. In the “Value” field, enter the first option. In this example, the color red.

2. The “Price modifier” and “Weight modifier” fields are used if one version of a product costs or weighs more than another version of a product. In this example, a red shirt doesn’t cost or weigh any more then another color shirt, so these field were left at 0.

3. Likewise, the “SKU modifier” field is used if a version of a product requires a unique SKU. In this example, it did not.



4. Click save. The option available for the parameter will be displayed.

5. To add more options, click on the “Add option” link again and repeat

steps 8-11.

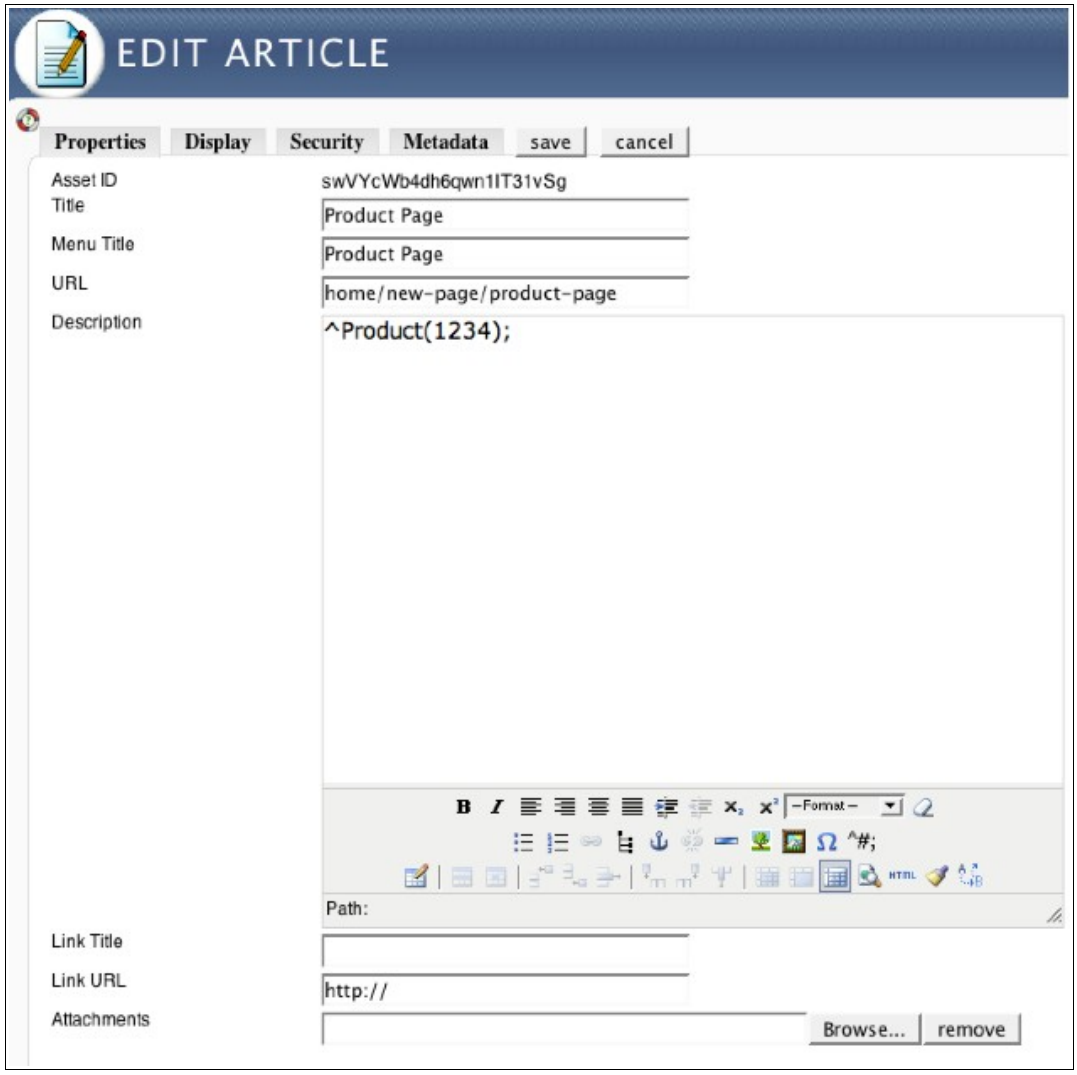
Display Products

Once your commerce system is set up and your products have been added, it's time to display your products on your site.

1. Begin on the page that will display your products (in this example, the Product Page).

2. In the New Content menu of the Admin Bar, select an asset that can display macros. In this example, an Article was used, but you could use any asset capable of displaying macros.

3. The Add Article screen will open.

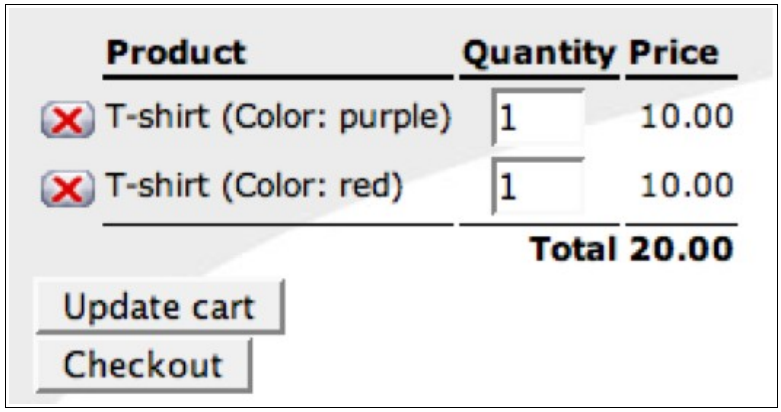


4. Enter a title for your product in the “Title” field.

5. In the “Description” field enter the product macro, which is ^Product(SKU);. Use the SKU entered at the time the product was added through the Product List screen. For this example you'd enter ^Product(1234);

6. Click save and the product will display on the page.

When users click on the “Add to cart” link next to the item they would like to purchase they will be shown their shopping cart, indicating the items chosen for purchase and their prices. The quantity field may be altered, the red X clicked to remove the product, or the Checkout button clicked to proceed with the purchase.



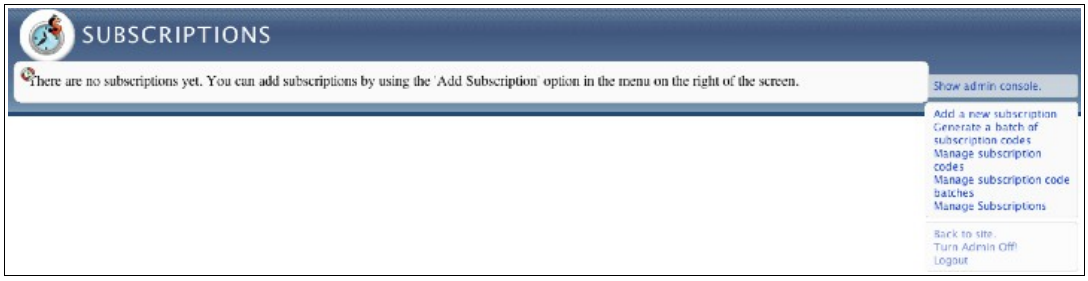
Subscriptions

Subscriptions are a great way to offer products or services on your site that otherwise would be unavailable to the average user. Setting up a subscription is a combination of commerce, user, and group set-up.

Before you begin, establish the group who will be allowed to access this subscription. You can refer to the Users and Groups chapter for full instructions on this. Also, set up your commerce system according to the directions in this chapter.

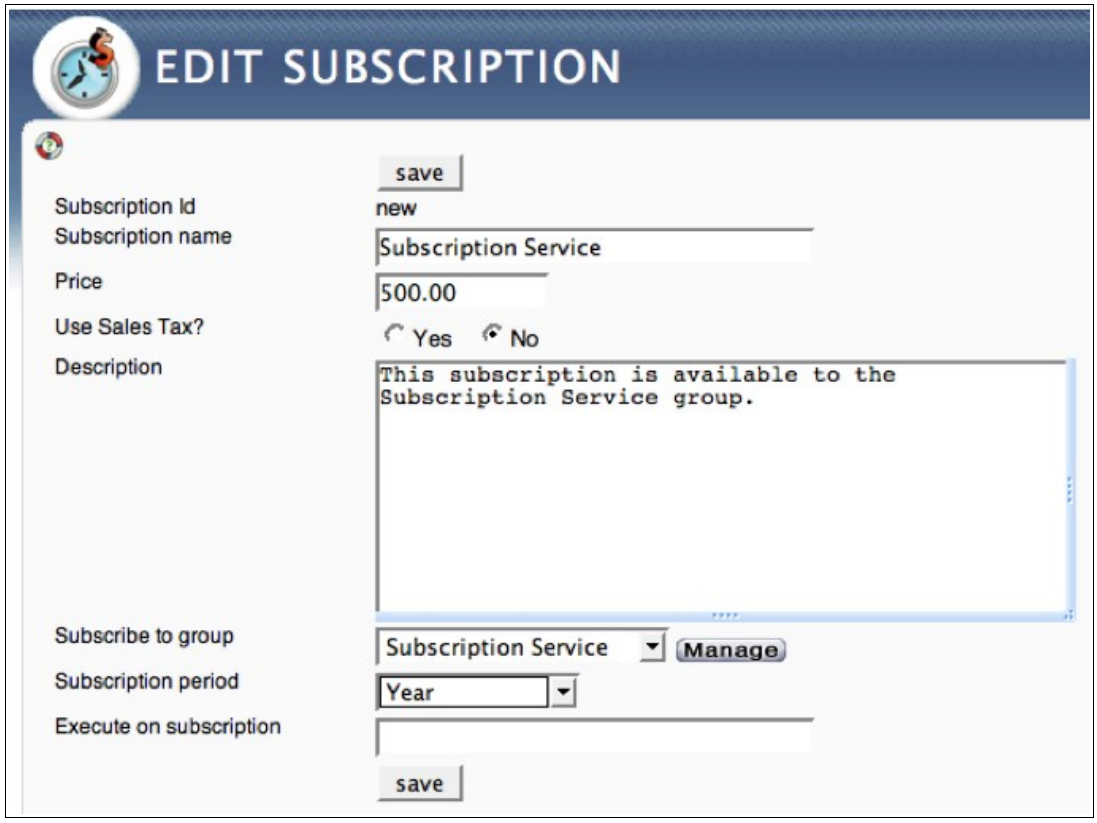
Set up the subscription through the Admin Console:

1. Select Subscription from the Admin Console. The "Subscriptions" screen will open. On this screen will be listed all the subscriptions currently available for purchase on your site; if none exist yet, none will be listed.



2. To add a new subscription, select the "Add a new subscription" link on the far right hand side of the screen.

3. The "Edit Subscription" screen will open. At the top of this screen is a Subscription Id field. This will be a unique identifier for this specific subscription. You will need this id later when displaying the subscription for sale.



4. Enter the name of the subscription in the“Subscription name” field.

5. Enter the price of the subscription in the “Price” field.

6. Designate if sales tax will be applied.

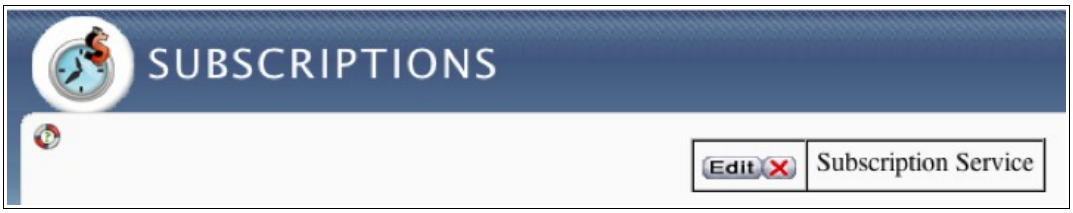
7. In the “Description” field, enter a brief description of the subscription. This can be an explanation of the subscription services or simply an indication of what group this subscription applies to.

8. In the “Subscribe to group” field select the group assigned to this subscription. This is the same group created in step one. Users who purchase this subscription will be added to this group upon purchase.

9. Indicate the length of this subscription in the “Subscription period” field.

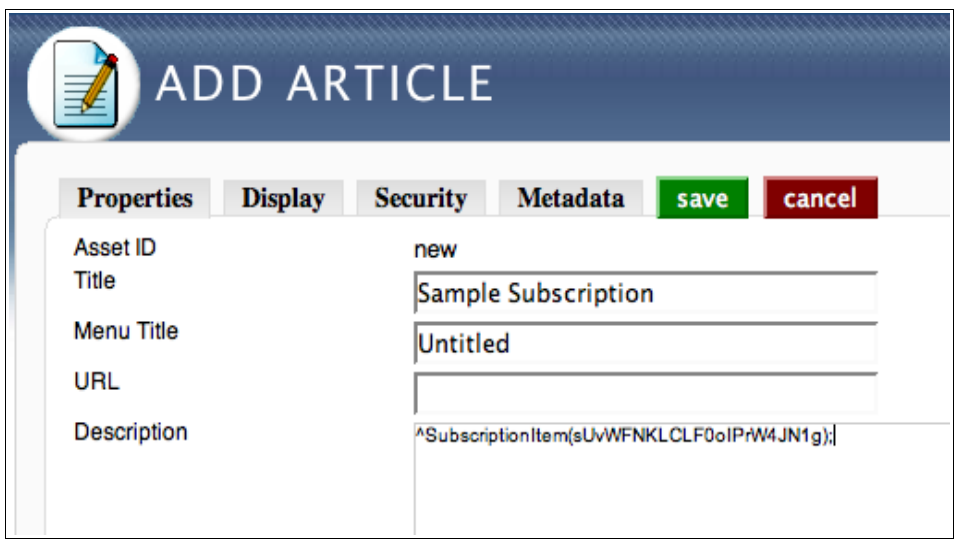
10. The “Execute on subscription” field is an area where a perl script can be entered to be called upon a user’s purchase of this subscription.

11. Click save. The subscription will be added to the Subscriptions screen. To delete a subscription from the system, click on the red X.



Subscriptions are displayed for sale in much the same way as a product, only you will use the Subscription macro: ^SubscriptionItem(SubscriptionID); You will need to copy the unique Subscription ID generated for this subscription from the Add/Edit Subscription screen to use in this macro.

Select an asset capable of displaying a macro, such as an article, and enter the subscription macro in the Description field.



Save to display the subscription for sale on your site.

Assets/pages that contain subscription services/products also need to be assigned the appropriate group to view access in the Security tab.

Secure Socket Layer (SSL)

At some point, you'll likely want to enable SSL (Secure Socket Layer) on your web site. You may want to just encrypt logins to protect your users' passwords, or you might want to protect site data and commerce functions. No matter your reasons, setting up SSL is very easy, and we recommend that you do it sooner rather than later.

What Is SSL?

Normal web traffic is sent unencrypted over the Internet. That is, anyone with access to the right tools can snoop all of that traffic. Obviously, this can lead to problems, especially where security and privacy is necessary, such as in credit card data and bank transactions. The Secure Socket Layer is used to encrypt the data stream between the web server and the web client (the browser).

SSL makes use of what is known as asymmetric cryptography, commonly referred to as public key cryptography (PKI). With public key cryptography, two keys are created, one public, one private. Anything encrypted with either key can only be decrypted with its corresponding key. If a message or data stream were encrypted with the server's private key, it can be decrypted only using its corresponding public key, ensuring that the data only could have come from the server.

If SSL utilizes public key cryptography to encrypt the data stream traveling over the Internet, why is a certificate necessary? The technical answer to that question is that a certificate is not really necessary - the data is secure and cannot easily be decrypted by a third party. However, certificates do serve a crucial role in the communication process. The certificate, signed by a trusted Certificate Authority (CA), ensures that the certificate holder is really who he claims to be. Without a trusted signed certificate, your data may be encrypted; however, the party you are communicating with may not be who you think. Without certificates, impersonation attacks would be much more common.

Enabling SSL

Follow these easy steps to enable SSL on your web server.

Step 1: Generate a Private Key

The openssl toolkit is used to generate an RSA Private Key and CSR (Certificate Signing Request). It can also be used to generate self-signed

These instructions assume you're using the WebGUI Runtime Environment, but will work equally well in a standard environment.

The first step is to create your RSA Private Key. This key is a 1024 bit RSA key which is encrypted using Triple-DES and stored in a PEM format so that it is readable as ASCII text.

```
mkdir -p /data/domains/www.example.com/certs
cd /data/domains/www.example.com/certs
. /data/wre/sbin/setenvironment.sh
openssl genrsa -des3 -out server.key 1024
```

The output will look something like this:

```
Generating RSA private key, 1024 bit long modulus
.....................................................++++++
.....++++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Step 2: Generate a Certificate Signing

Plain Black offers discount certificates. Check out webguistore.com for details.

Once the private key is generated, a Certificate Signing Request can be generated. The CSR is then used in one of two ways. Ideally, the CSR will be sent to a Certificate Authority, such as Thawte or Verisign, who will verify the identity of the requester and issue a signed certificate. The second option is to self-sign the CSR, which will be demonstrated in *Step 4: Generating a Self-Signed Certificate*.

During the generation of the CSR, you will be prompted for several pieces of information. These are the X.509 attributes of the certificate. One of the prompts will be for "Common Name". It is important that this field be filled in with the fully qualified domain name of the server to be protected by SSL. If the website to be protected will be http://www.example.com, then

enter www.example.com at this prompt. The command to generate the CSR is as follows:

```
openssl req -new -key server.key -out server.csr
```

The output will look something like this:

```
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:WI
Locality Name (eg, city) [Newbury]:Madison
Organization Name (eg, company) [My Company Ltd]:Example Co.
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) []:www.example.com
Email Address []:info@example.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Step 3: Remove Passphrase from Key

One unfortunate side effect of the pass-phrased private key is that Apache will ask for the pass-phrase each time the web server is started. Obviously, this is not necessarily convenient as someone will not always be around to type in the pass-phrase, such as after a reboot or crash. mod_ssl includes the ability to use an external program in place of the built-in pass-phrase dialog; however, this is not necessarily the most secure option either. It is possible to remove the Triple-DES encryption from the key, thereby no longer needing to type in a pass-phrase. If the private key is no longer encrypted, it is critical that this file only be readable by the root user! If your system is ever compromised and a third party obtains your unencrypted private key, the corresponding certificate will need to be revoked. With that being said, use the following command to remove the pass-phrase from the key:

```
cp server.key server.key.org

openssl rsa -in server.key.org -out server.key

chmod 700 server.key
```

The newly created server.key file has no more passphrase in it. Your *certs* folder should have three files in it similar to this, with these permissions:

```
-rw-r--r-- 1 root root 745 Jun 29 12:19 server.csr
-rw------- 1 root root 891 Jun 29 13:22 server.key
-rw-r--r-- 1 root root 963 Jun 29 13:22 server.key.org
```

Step 4: Generating a Self-Signed Certificate

If you don't plan on having your certificate signed by a CA, or you wish to test your new SSL implementation while waiting for your CA to sign your certificate, you should generate a self signed certificate. If you are having a CA sign your certificate, and you don't need to test SSL right now you can skip this step, and just wait for your certificate to come back from your CA. When you get it, put it in a file called *server.crt*.

Though you can technically generate a certificate that will last longer than 365 days, it's not recommended. The shorter the interval between generating certificates, the less likely you are to be the victim of crack attacks.

The self-signed certificate will generate an error in the client browser to the effect that the signing certificate authority is unknown and not trusted.

To generate a self-signed certificate which is good for 365 days, issue the following command:

```
openssl x509 -req -days 365 -in server.csr \-signkey server.key -out server.crt
```

```
Signature ok
subject=/C=US/ST=WI/L=Madison/O=Example Co./OU=Information
Technology/CN=www.example.com/Email=info@example.com
Getting Private key
```

Step 5: Configuring SSL Enabled Virtual Hosts

Now that you've got a signed certificate, all you need to do is tell Apache that you want to use it. Create an SSL virtual host in your site's mod_proxy config file. Edit www.example.com.modproxy and add this to the bottom of the file:

```
Listen 10.0.0.1:443
<VirtualHost 10.0.0.1:443>
DocumentRoot /data/domains/www.example.com/public
ServerName www.example.com
ServerAlias example.com
```

```
SSLEngine on
SSLCertificateFile /data/domains/www.example.com/certs/server.crt
SSLCertificateKeyFile /data/domains/www.example.com/certs/server.key
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
CustomLog /data/domains/www.example.com/logs/access.log "%t %h
%{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

You need a unique IP address for each SSL host you serve.

Step 6: Restart Apache and Test

Now all you need to do is restart your web server and test.

```
rc.webgui restartmodproxy
```

Now visit your web site and test it at https://www.example.com

Logging

WebGUI has a very powerful logging system that enables you, the Administrator, to get exactly the information you need to keep WebGUI running smoothly and error free. This chapter covers some of the ways in which you can harness the power of the WebGUI logging system.

Basic Configuration

All WebGUI logging is controlled through a single configuration file. This configuration file starts out very small and simple, but can grow to become very complex depending upon your needs.

The log file is located in /data/WebGUI/etc and is called log.conf. You have to create this file as part of your WebGUI install, or if you're using the WRE, it will create the log.conf file for you. If you need to create this file, the easiest way to do it is to copy log.conf.original to log.conf.

This is what the default log file looks like:

```
log4perl.logger = WARN, mainlog

log4perl.appender.mainlog = Log::Log4perl::Appender::File

log4perl.appender.mainlog.filename = /var/log/webgui.log

log4perl.appender.mainlog.layout = PatternLayout

log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

This can be a bit overwhelming to look at all at once, so this example breaks it down into its parts, line by line.

The first line tells the logging system that you're going to create a logging object called "mainlog" and that it's going to log "WARN", "ERROR", and "FATAL" messages. Have a look atthat line:

Note that from now on the "mainlog" alias is used in each configuration line, which tells the logging system which logger these directives apply to. This is important because it's what allows you to define multiple loggers, each with different behaviors.

```
log4perl.logger = WARN, mainlog
```

The second line invokes the “appender”, or mechanism that you’re going to use to write the log out somewhere. By default, it is usually good to put logs into files. However, you can put logs out to databases, email addresses, logging servers, or all sorts of other places. In this case, the appender used will put the logging information into a file. Take a look:

```
log4perl.appender.mainlog = Log::Log4perl::Appender::File
```

The third line tells the appender where you’re going to store the log. Given that the appender you’re using writes out to a file, you give it a file path. If the appender used was an email based appender, you’d probably specify an email address and mail server configuration. The default location for the WebGUI log is /var/log/webgui.log. Take a look at the line that sets this up:

```
log4perl.appender.mainlog.filename = /var/log/webgui.log
```

The fourth line tells the appender how you want to format the information that the logger is given. There are various logging standards for layout, and you could use one of them, but you want to grant yourself maximum control, so tell the appender that you wish to use a pattern based layout. Take a look at that line:

```
log4perl.appender.mainlog.layout = PatternLayout
```

The fifth, and final, line specifies the pattern that you want to use to lay out each line in the log. There are lots of symbols used here, and each represents a piece of data. We’ll get into what each means later, but for now take a look at the line that defines the pattern:

```
log4perl.appender.mainlog.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Log Levels

WebGUI has five levels of logging, and each adds its own distinct kind of information to the log. They are FATAL, ERROR, WARN, INFO, andDEBUG. Each one builds on the previous, so ERROR includes FATAL, and WARN includes ERROR and FATAL, and so on.

FATAL

The fatal log level only logs “show-stopper” errors. In other words, errors that are so devastating that WebGUI cannot continue processing the page because doing so is either impossible or would be dangerous. Thankfully,

the number of things that can cause fatal errors is pretty low. They generally either indicate that WebGUI cannot connect to some critical resource, such as the database or file system, or that a programmer has introduced a bug into the system. You should never see fatal errors.

ERROR

The error log level logs all errors that do not fall under the fatal category. These are breaks in functionality, but WebGUI can continue processing the page. Examples are that WebGUI can't overwrite a file that is write protected, it tries to connect to an external server, but the connection is refused, or perhaps WebGUI detected that a user has entered invalid data into a form. Most things that fall under this category are correctable through configuration changes, or other small adjustments.

WARN

The warn log level advises you of warnings, or things that aren't errors, but could be harmful none-the-less. Examples are when a user tries to do something s/he doesn't have the privileges to do, or when site visitors are requesting pages that don't exist.

INFO

The info log level gives you some insight into the inner workings of WebGUI. You'll generally see these when WebGUI is working on multi-step complex process, such as processing workflows. This log level can be extremely useful when you're trying to troubleshoot a problem with the system.

DEBUG

The debug log level is a very verbose log level that tells you everything that's going on in the internals of WebGUI. It shows you stack traces for the errors and fatals, it shows you every database query running through the system, and much more. In fact, from an administrator's point of view there's really almost **too much** information being provided at the debug level. Unless you are a developer, or have become very familiar with WebGUI, this log level will likely only serve to confuse you rather than help you diagnose a problem.

Separate Log Files Per Site

Creating a separate log file for each of your sites is quite simple. It's really just a variation on the default logger configuration that comes with WebGUI. Let's say that you have two sites, www.foo.com and www.bar.com, and you want a separate log file for each. Furthermore, you want Spectre (the workflow controller) to log its messages in its own file as well.

```
log4perl.logger.www.foo.com.conf = WARN, foo

log4perl.appender.foo = Log::Log4perl::Appender::File

log4perl.appender.foo.filename = /var/log/webgui.foo.log

log4perl.appender.foo.layout = PatternLayout

log4perl.appender.foo.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n


log4perl.logger.www.bar.com.conf = WARN, bar

log4perl.appender.bar = Log::Log4perl::Appender::File

log4perl.appender.bar.filename = /var/log/webgui.bar.log

log4perl.appender.bar.layout = PatternLayout

log4perl.appender.bar.layout.ConversionPattern = %d - %p - %c - %M[%L]- %m%n


log4perl.logger.spectre.conf = WARN, spectre

log4perl.appender.spectre = Log::Log4perl::Appender::File

log4perl.appender.spectre.filename = /var/log/spectre.log

log4perl.appender.spectre.layout = PatternLayout

log4perl.appender.spectre.layout.ConversionPattern = %d - %p - %c - %M[%L] - %m%n
```

Notice that you can reference each site (and Spectre), by its config file name.

```
log4perl.logger.www.bar.com.conf = WARN, bar
```

And you give each logger object its own unique identifier:

```
log4perl.logger.www.bar.com.conf = WARN,bar
```

Sending Emails From Logs

One way to make absolutely sure that your WebGUI system is operating in good health is to have the system email you on all fatal errors. Some people go so far as to email themselves on all errors and warnings, but depending on how your users use your site, that can fill up your inbox quite quickly, and be relatively expensive performance-wise.

To send yourself emails directly from the WebGUI log, you first need to install a new perl module called *Log::Dispatch::Email::MailSend*. Here are the commands to do this from the command line.

```
. /data/wre/sbin/setenvironment

cpan
```

Once in the CPAN interactive shell type:

```
install Log::Dispatch::Email::MailSend

exit
```

Now that you've installed this module, all you have to do is add a new logger object to your log.conf file, and restart Apache. The new logger object will look similar to the following, be sure to replace this example with your own information where you see bold lettering:

```
log4perl.category = FATAL, MailMElog4perl.appender.MailME =
Log::Dispatch::Email::Sendlog4perl.appender.MailME.to = info@example.com
log4perl.appender.MailME.subject = Don't panic but we've got a FATALerror
log4perl.appender.MailME.layout = SimpleLayoutlog4perl.appender.MailME.buffered = 0
```

You're now getting emails about fatal errors. Note that these messages are small enough that they can work with email-to-text pagers (like your cell phone).

Troubleshooting

If you're reading this chapter then you're either being a good administrator, and preparing for the future, or something bad has just happened and you're trying to figure out what you can do about it. Either way, this chapter should give you some good tips about how to proceed.

Having said that though, there's no way to give you a comprehensive guide that would cover all things. If you really get stuck, don't spin your wheels, give Plain Black support a shot, and get your site back online sooner rather than later.

Don't Panic

The worst thing you can do in an emergency situation is panic. This is true of any emergency, whether it be a down server, you're trapped on a deserted island, or your wife is having a baby.

Collect Your Thoughts And Gather Intel

Breathe. Now that you're breathing again, find out what the symptoms of the problem are. Not just "we're down", but perhaps when we went down, what was going on with the server at that time (perhaps backups or web stats generation).

Make Notes, Not Assumptions

As you make discoveries, make notes of those discoveries so they can help with your diagnosis of the problem, and your post-mortem to stop the problem from happening again in the future. **Do not** make assumptions at this point. Assumptions will likely take you down the wrong path and make you waste hours of valuable time. More importantly, assumptions might make you miss the actual cause when it passes in front of you.

Reading Logs

If you're having a problem, the first place you should go is directly to your log files. Look at your WebGUI logs, your Apache error logs, your MySQL logs, and sometimes even your operating system logs will have some insight for you. Sometimes, the logs will tell you exactly what's wrong: "Your out of disk space", "Can't connect to the mail server", etc. Most of

the time, the log file will just be giving you your first round of symptoms.

Don't assume the symptom is the problem. For example, if the log tells you that it can't connect to the mail server, does that mean the mail server is down? Does that mean that WebGUI has a bug? Does that mean someone made a firewall change? Does that mean your router has failed? Maybe all of the above? If you get into this mindset, you're doomed to failure. After you have your list of symptoms, you need to ask yourself only one question: **What has changed?**

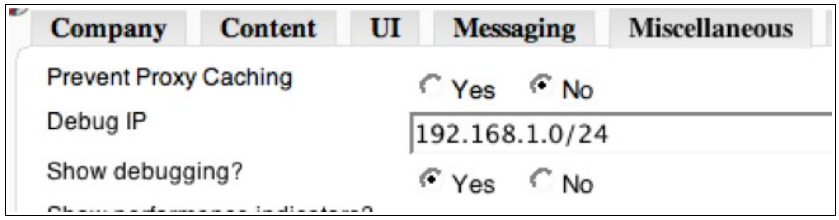
Everything was working until it went down, right? So what has changed? Maybe it's none of the above things, but rather a user went into the settings and changed the mail server from yours to some other mail server and that's the mail server it can't contact. But, if you made the assumption that it was your mail server, you could spend decades trying to troubleshoot a problem that's not there.

If you find that the WebGUI error log is reporting an error, but it's not enough for you to go on, perhaps you should increase your log level. Doing so may just give you that added bit of detail that you need to see what's going on. See the chapter called "Logging" for details.

WebGUI Debug Mode

WebGUI debug mode is most useful for developers, but sometimes it can also help in diagnosing problems with WebGUI. To turn it on, follow these steps:

1. Log in and turn admin on.
2. Go to the Admin Console tab of the Admin Bar and then choose Settings.
3. Flip to the "Miscellaneous" tab.
4. Fill in your IP address or your entire subnet in CIDR notation in the "Debug IP" field.
5. Then set "Show debugging?" to yes.
6. Click on the "save" button.



Now as you browse pages you will see debugging output at the bottom of every page as long as your IP address matches the Debug IP field in the settings. However, all your users will just see the pages as normal.

This can be especially useful if you believe that certain actions, pages, assets, or macros are causing errors on your site. Then, all you have to do is visit the problematic page, or perform the problematic action and see if there are any error messages in the debug output that can help you resolve the problem.

Spectre Debug Mode

Spectre has a debug mode that can help you work through connectivity issues between it and WebGUI. To start it running in debug mode do the following:

```
cd /data/WebGUI/sbin

perl spectre.pl --run --debug
```

Now Spectre will inform you of everything it's doing when it does it. You'll be able to see if it's able to connect to WebGUI, if it's getting error messages, how many messages are in its queue, and a whole host of other information.

Spectre's Automated Tests

Spectre also has a test mode that can be used to test whether it can successfully connect to WebGUI. While Spectre is running (whether in debug mode or not), run the following commands:

```
cd /data/WebGUI/sbin

perl spectre.pl --test
```

It should report that it can connect just fine to everything like this:

```
Running connectivity tests.
Testing www.example.com.conf
Tests completed.
```

However, if you get back something like this:

```
Running connectivity tests.
Testing www.example.com.conf
ERROR: Couldn't connect to WebGUI site www.example.com.conf
Tests completed.
```

Then you've got a connectivity problem. It could be that either Spectre or WebGUI is offline, or it could be that one or the other is misconfigured.

One of the most common mistakes people make is in setting the *spectreSubnets* directive in each of their WebGUI config files. It needs to contain all the IP addresses that Spectre may use to communicate to the server. Here's an example where the server has two IP addresses assigned to it:

```
"spectreSubnets" : ["127.0.0.1/32", "111.111.111.111/32", "222.222.222.222/32"],
```

Notice that the loopback IP address is also included. This is important. Note that it's not possible to simply set the list to 0.0.0.0/0 (ie: the whole Internet) for security reasons.

Spectre's Runtime Status

Spectre has a runtime status which can go a long way toward troubleshooting problems with workflows. To view status of workflows inside of Spectre run the following commands:

```
cd /data/WebGUI/sbin

perl spectre.pl --status
```

If Spectre has no workflows waiting to run, then the output will look like this:

```
Suspended Workflows    0
    Waiting Workflows    9
    Running Workflows    0
        Total Workflows    9
```

Otherwise you can expect output more like this (rotated to fit):

```
Workflows    1
     Priority  Sitename           Instance Id             Last State   Last Run Time
      11/10  www.plainblack.com  97jenYkHWA7g0OqOJWdsCw  comm error   Mon Feb 19 17:05:19
 Waiting Workflows    6
     Priority  Sitename           Instance Id             Last State   Last Run Time
      11/10  www.plainblack.com  icKhsoSbuK-AwBdJ38tIjg  complete     Mon Feb 19 17:05:22
      11/10  www.plainblack.com  pEcyYg4cYy0kX_8wfeuihQ  complete     Mon Feb 19 17:05:23
      11/10  www.plainblack.com  -pzElVJUN9JX__4t-NleVA  complete     Mon Feb 19 17:05:24
      11/10  www.plainblack.com  HsMIaMCyoHl_xSJo2b9XkQ  complete     Mon Feb 19 17:05:25
      11/10  www.plainblack.com  XErByV4zt9qR6QnB2qdPNw  complete     Mon Feb 19 17:05:26
      20/20  www.plainblack.com  uuzbZrghYwiSxEe9fou3mQ  never run

 Running Workflows    3
     Priority  Sitename           Instance Id             Last State   Last Run Time
      11/10  www.plainblack.com  LPYhrA-eKl1H_kgLwQ833Q  complete     Mon Feb 19 17:05:20
      11/10  www.plainblack.com  No9wiI6AgJf2aSYSjL_txw  complete     Mon Feb 19 17:05:22
   Total Workflows    9
```

WebGUI / Spectre Communication

If you're having WebGUI / Spectre communication problems, you can manually run the transactions from Spectre to WebGUI to see if there's anything wrong.

You'll need a program called cURL to do this. It comes with most Unix, Linux, and BSD operating systems, and the Windows WRE. You'll also need to do this from the command line of the machine that runs Spectre.

The first transaction you can try is the one Spectre calls on each site to get that site's configuration data.

```
curl http://www.example.com/?op=spectreGetSiteData
```

It should return a JSON object with all the Workflow Instances and Scheduler Tasks that need to be run.

```
{"workflow":[],"cron":[{"sitename":"www.example.com","taskId":"pbcron000000000000001","dayOfMonth":"*","dayOfWeek":"*","gateway":"/","minuteOfHour":"30","hourOfDay":"23","cookieName":"wgSession","runOnce":"0","monthOfYear":"*"},{"sitename":"dev.localhost.localdomain","taskId":"pbcron0000000000000002","dayOfMonth":"*","dayOfWeek":"0","gateway":"/","minuteOfHour":"30","hourOfDay":"1","cookieName":"wgSession","runOnce":"0","monthOfYear":"*"},{"sitename":"dev.localhost.localdomain","taskId":"pbcron0000000000000003","dayOfMonth":"*","dayOfWeek":"*","gateway":"/","minuteOfHour":"15","hourOfDay":"*","cookieName":"wgSession","runOnce":"0","monthOfYear":"*"},{"sitename":"dev.localhost.localdomain","taskId":"pbcron0000000000000004","dayOfMonth":"*","dayOfWeek":"*","gateway":"/","minuteOfHour":"*/5","hourOfDay":"*","cookieName":"wgSession","runOnce":"0","monthOfYear":"*"}]}
```

The next transaction you can try is the one that runs a workflow activity. Note the \ before the ; in the following command. That's because on most systems the semi-colon separates multiple commands on a single line. By escaping it with a backslash it's saying it wants to pass the semi-colon along through the URL.

```
curl http://www.example.com/?op=runWorkflow\;instanceId=xxxxxxxxxxxxxxxxxxxxxx
```

It should return a one word status.

```
complete
```

The final transaction you can try is the one that runs a scheduler task. Again, you escape the semi-colon with a backslash.

```
curl http://www.example.com/?op=runCronJob\;taskId=xxxxxxxxxxxxxxxxxxxxxx
```

Again, a one word status should be returned.

```
done
```

WebGUI Advisories

If you're having a problem and you can't figure out how to fix it, perhaps it's a bug and it's been fixed already. As a WebGUI administrator, it's your job to stay on top of the releases of WebGUI, and upgrade to get bugs fixed, as well as apply patches for security problems. Luckily, we make this easy for you. You can subscribe to our advisories list at www.getwebgui.com. This is a very low traffic list, and is only posted to when a new release of WebGUI or the WRE comes out, or if a serious security problem was found. We strongly encourage all WebGUI administrators to be subscribed to this list, or at the very least check it out once a week.

Reporting Bugs

So you've tried everything and you're sure you have a bug on your hands? Before you report it, create a demo out on http://demo.webgui.org and see if you can reproduce the bug there. If you can, you probably have a bug; otherwise, you probably have a local configuration issue of some sort. If you think it's a bug, read on.

Stop! Before you report a bug, you need to know what the difference between a bug and a request for enhancement (RFE) is. Many people confuse the two.

A bug is a functional problem with something not working as advertised. For example, if you see a "save" button, and click on it, and it does not save the data in the form, then that is a bug. However, if the save button is saving the data, but you wish that after saving it would take you to some other page, then that's an RFE. The reason is that the save button is working, you just want it to do something beyond what it's designed to do. If this is the case, go to the "Making a Request For Enhancement" section later in this chapter.

Reporting a bug is very easy. You can simply go to http://webgui.org/bugs and submit your bug. However, reporting a bug correctly is not as easy. First, you need to check if the bug you're reporting has already been reported. Second, if you want the bug fixed expediently, and to your satisfaction, help us help you. You can do that by including lots of information in your bug report. The following things should be in every bug report:

1. Your WebGUI version number.
2. Your WRE version number (or if you're not using the WRE, then provide us with information about what platform you are using).
3. A step by step account of how to reproduce the bug.
4. A description (or screenshot) of what is happening.
5. A description of what you think should be happening.

Also, with each bug report, include anything else that you think may help us diagnose the bug. And if you're the sort that can fix bugs, feel free to include a suggested patch to resolve the problem.

Making a Request For Enhancement

A Request For Enhancement (RFE) is any change in functionality you want made for WebGUI. It could be that you want a new asset written that will handle the publishing of __insert__some__custom__thing__here__, or maybe you want to change some existing functionality to work slightly differently. Whatever the case, you should publish an RFE.

To publish an RFE, simply go to http://webgui.org/rfe and type out what you want done.

Before you publish your RFE, though, see if anyone else has already published something similar. If they have, you should publish your comments as a reply to theirs, and then apply your karma to the post to move it up to the top of the list.

When you do finally start publishing your RFE, be sure to be detailed. If you have an idea about how something should look, consider attaching a screenshot. If you know how to write HTML, CSS, Javascript, Perl, SQL, or any other code/script that may provide additional insight into your request, include that as well. And most importantly, include the thing that everyone forgets, include why this is such an important feature.

More Resources

If you didn't get enough information in this book, never fear, there are many other sources of information about WebGUI out there.

Commercial Offerings

Plain Black Corporation provides a full line of commercial products and services to ensure your success with WebGUI. There are several other books just like this one available with other audience focuses. We also provide training, hosting, professional services, and a whole line of support options. Check out all these offerings at: **www.webguistore.com** or **www.plainblack.com**

Web Sites

There are lots of free web resources out there to help you achieve success. Here is a short list of web sites that are there to help:

www.webgui.org - Get all sorts of information about what WebGUI is, why everyone should be using it, and access community support.

www.spreadwebgui.com - Let the world know why WebGUI is what they should be using. Also see what other users have done to get the word out.

www.getwebgui.com - Download the latest version of WebGUI, the WebGUI Runtime Environment, and get the latest security advisories.

update.webgui.org – Browse the Plain Black download mirror of WebGUI directly to find old versions of WebGUI.

wiki.webgui.org – The place to find community documentation and tips.

svn.webgui.org - Browse the WebGUI code repository to see what new features are being added and what bugs have been fixed before the official release comes out.

www.webguiworldwide.org – A list of all the known WebGUI related sites around the world.

translate.webgui.org - Help work on WebGUI user interface translations, and get the latest translations before they are made publicly available. Also

known as i18n.webgui.org.

demo.webgui.org - Try out the latest beta of WebGUI, try to replicate a bug on a known working system, or just play around with a feature you're not sure enough of in a failsafe play area.

IRC (Internet Relay Chat)

If you like instant gratification, join our IRC channel at:

url: irc://irc.freenode.net#webgui

server: irc.freenode.net

room: webgui

Join us on the channel, get some advice, and then stick around to help others out with your new found knowledge. That's what community is all about!